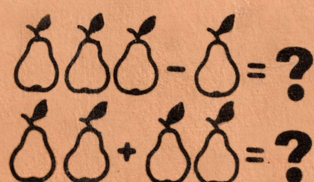
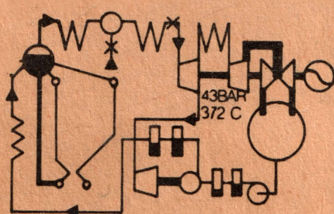
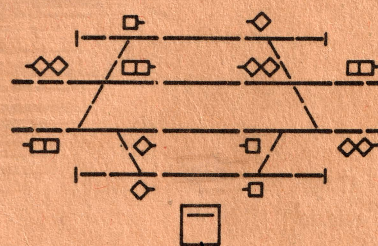


AUTOMAT PROGRAMABIL PENTRU INSTRUIRE









manual de utilizare

INTREPRINDEREA ELECTRONICA

CALCULATOR DE INSTRUIRE PERSONAL - 02

- manual de utilizare -

1989

Elaboratori:

Ing. VICTOR POPESCU - CEPECA

.coordonare generala
.capitolele 1-9

Mat. LAZAR TRIFAN - I. Electronica

.capitolele 10 si 11

Ing. OVIDIU RANCU - I. Electronica

.capitolul 12

Ing. I. ROSCA, GH. BOGUS, V. TRIFESCU, A. PUSCASU - I. Electronica

.anexele D si E

Ing. DAN GHEGHEA si Designer EDUARD MILEA - I. Electronica

.caseta demonstrativa

CUPRINS

INTRODUCERE

Stimate cititor,

Felicitari pentru ca v-ati hotarit sa folositi "Automatul Programabil pentru Instruire" care, va asiguram de la inceput , va va oferi clipe de placere si satisfactie.

Automatul programabil pentru instruire personala CIP a fost conceput si va este oferit ca un sprijin de nadejde in activitatea de instruire scolara, in aplicatii practice din activitatea profesionala, in industrie, in medicina, in cercetare, in calcule economice.

CIP-ul a fost astfel realizat incit sa inteleaga si sa execute programe concepute de firme renumite pentru calculatoarele de larga raspindire SINCLAIR-SPECTRUM.

De asemenea, CIP-ul accepta programe elaborate pe calculatoarele romanesti HC-85, TIM-S si COBRA.

CONDITII DE GARANTIE

CIP este garantat de I. Electronica 12 luni de la data cumpararii.

Ies din garantie aparatele la care a fost violat sigiliul, sau a caror reparare a fost incredintata unor persoane neautorizate de catre unitatea de depanare, unitate indicata in certificatul de garantie de magazinul de la care ati cumparat produsul.

Intretinerea aparatelor in perioada de garantie, service-ul, se asigura de catre Fabrica Service a I. Electronica prin unitatile sale de lucru din intreaga tara inclusiv din Municipiul Bucuresti (nominalizate in reteaua unitatilor service - AX 1443).

CARACTERISTICI TEHNICE

-microprocesor Z80A (MMN 80 CPU)
-memorie EPROM -2Ko
-memorie RAM -64Ko
-tastatura QWERTY cu 40 taste
-afisare -televizor alb-negru sau color PAL (canalul 8 OIRT)
          -monitor alb-negru sau color
-opt culori cu doua trepte de luminozitate
-rezolutie - 192 x 256 in regim grafic
           - 24 x 32 in regim alfanumeric
-limbaj - BASIC-S
-alimentarea 220 Vca/5 Vcc
-puterea consumata max 20 VA
-gabarit 330 x 280 x 60 (mm)
-greutate cca 3,5 Kg cu sursa de alimentare.

INVENTAR DE LIVRARE

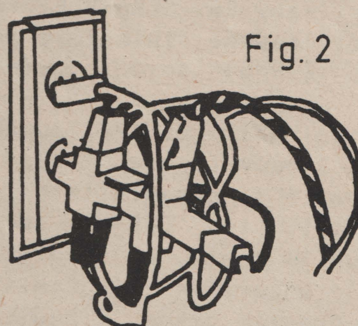
La cumparare, in ambalajul produsului trebuie sa gasiti:
-unitatea de baza CIP;
-sursa de alimentare,
-caseta demonstrativa;
-cablu de conectare la TV;   tr sa gasiti:
-manual de utilizare;
-schema electrica;
-reteaua unitatilor service;
-certificat de garantie;

-doua siguranțe de rezerva;
-mufa DIN tata cu sase contacte.

STRICT INTERZIS

-Sa folositi cordoane de retea cu izolatia compromisa (fig. 1).


Fig. 1


Fig. 2

-Sa introduceti doua triplustechere in aceeasi priza (fig. 2).
-Sa inlocuiti sigurantele cu alte valori decit cele prevazute in schema aparatului.
-Sa folositi sursa de alimentare cu capacul desfacut, intrucit riscati sa va electrocutati.

RETINETI!

1. Nu introduceti jack-ul de alimentare in automatul programabil decit cu sursa neconectata la retea.
2. Nu astupati orificiile de aerisire ale sursei de alimentare si ale automatului programabil in timpul functionarii.
3. Nu introduceti sursa de alimentare in spatii inchise, in timpul functionarii.
4. Nu instalati CIP-ul si sursa de alimentare linga vase cu lichid (vase cu flori, spre exemplu) intrucit prin rasturnarea lor accidentala lichidul poate patrunde in interior provocind avarii sau incendii.
5. In caz ca CIP-ul si sursa de alimentare au fost pastrate in conditii de umiditate ridicata sau a patruns accidental lichid in interior nu le porniti inainte de 12 ore.
6. Nu lasati aparatul sa functioneze nesupravegheat.

NEGLIJAREA ACESTOR REGULI POATE DUCE LA SUPRAINCALZIREA PRODUSULUI SAU CHIAR LA INCENDIEREA LUI.

Acest manual va face cunostinta cu o noua creatie a electronicii si informaticii romanesti, prin unitatile lor reprezentative Intreprinderea Electronica si Institutul de Tehnica de Calcul si Informatica.

De ce se numeste CIP?

Aveti deci in fata un aparat electronic programabil denumit mai scurt si mai nostim "CIP", conceput pentru a fi folosit in procesul de invatamint elementar, mediu si superior.
In invatamintul prescolar pe CIP se pot prezenta notiuni elementare de grafica, muzica, desene animate, etc.
In invatamintul liceal aparatul poate fi utilizat eficient in instruire asistata, pentru insusirea materiilor.
In invatamintul superior CIP-ul poate reduce timpul necesar muncii de

rutina in realizarea proiectelor .

Automatul programabil de instruire il puteti utiliza pentru insusirea unor limbaje evoluate de programare a calculatoarelor.

Pentru cei "mai seriosi" dintre dumneavoastra, CIP-ul va permite sa il utilizati in aplicatii practice din activitatea profesionala, in industrie, in medicina, in cercetare, in calcule economice.

CIP-ul este capabil de multe alte utilizari. Cu el puteti juca pe ecranul unui televizor: sah, GO, SCRABBLE, fotbal, baschet, puteti invata sa conduceti masina, puteti desena in culori pe ecranul televizorului, puteti compune muzica etc., CIP-ul dovedindu-se un partener inteligent, rabdator si perspicace.

Dar pentru unii dintre dumneavoastra, poate mai interesanta decit folosirea programelor facute de altii va fi realizarea unor aplicatii proprii, in care imaginatia si creativitatea de care veti da dovada va vor face sa fiti foarte incintati, iar CIP-ul se va dovedi colaboratorul dumneavoastra inteligent, bun la toate, cu conditia sa fie invatat ce are de facut.

Desi poate uneori veti fi uimit de ce este capabil, retineti ca tot ce stie el sint creatii umane, CIP-ul fiind numai un automat programabil (larg cunoscut sub denumirea de calculator personal familial).

Ce fel de calculatoare personale exista?

Calculatoarele personale sint considerate ca fiind de doua mari categorii:

-calculatoare personale familiale (HOME COMPUTER), cu un pret accesibil pentru a putea fi cumparate "pentru acasa" si avind destule posibilitati de utilizare pentru calcule, desene, instruire, jocuri, etc.

-calculatoare personale profesionale (PERSONAL COMPUTER) cu posibilitati foarte mari de utilizare in institutii sau intreprinderi pentru aplicatii de cercetare-proiectare, activitati de birou, instruire asistata de calculator etc.

Dvs. ce aveti in fata?

Obiectul minunat cu care doriti sa lucrati, face parte din categoria calculatoarelor familiale si veti constata cu placere cit de mari ii sint posibilitatile de utilizare.

Ce programe pot fi folosite?

CIP-ul a fost astfel conceput incit sa inteleaga si sa execute programe facute de firme renumite pentru calculatoarele de larga raspindire SINCLAIR-SPECTRUM.

De asemenea, CIP-ul accepta programe elaborate pe calculatoare romanesti HC-85 si TIM-S.

Reciproc, programele realizate pe CIP vor putea fi executate pe calculatoare SINCLAIR ZX SPECTRUM sau pe calculatoarele romanesti HC-85 si TIM-S.

Cum se foloseste acest manual?

Acest manual va prezinta, simplu si practic, cunostintele necesare pentru a intelege ce cuprinde un automatul programabil cum functioneaza si cum puteti lucra curent cu el.

Pentru a intelege fara greutati deosebite ce vi se prezinta in continuare, este necesar sa il aveti la dispozitie astfel incit parcurgerea materialului sa o faceti simultan cu folosirea CIP-ului.

Veti avea foarte multe exemple, nu ezitati sa le verificati pe calculator. Veti constata ca limbajul utilizat este foarte exigent si

prietenos in acelasi timp, CIP-ul semnalindu-va orice abatere (eroare) si neexecutind decit comenzile corecte.

CIP-ul va ajuta sa va cunoasteti!

De fapt, automatul programabil este o oglinda a intelectului dvs., executind prompt cele ce-i veti da corect si clar, dar sanctionind si orice lipsa de atentie, de logica sau de cunoastere.

Manualul se adreseaza tuturor, fara a cere o pregatire prealabila in electronica sau informatica. El nu va arata cum se construieste un automat programabil, ci din ce este format, cum lucreaza si cum il puteti utiliza pentru diverse aplicatii.

Fiecare capitol este structurat astfel:

. aplicatie practica
. notiuni de baza
. exemple de programe
. intrebari recapitulative si exercitii
. probleme propuse pentru rezolvare
. raspunsuri

Acolo unde in text aveti lasat spatiu liber subliniat, va solicitam sa introduceti un raspuns al dvs. bazat pe notiuni explicate anterior.

Fiind un manual de autoinstruire, nimic nu va forteaza sa-l studiati contra cronometru. Apreciati posibilitatile dvs. de asimilare, care pot varia de la o zi la alta. Va indicam sa nu studiati mai putin de o ora, dar nici mai mult de trei-patru ore pe zi.

Materialul este astfel redactat incit sa asimilati cunostinte noi pas cu pas, cu minimum de efort, incit veti fi placut impresionat de posibilitatile dvs.

S U C C E S !!!

CAPITOLUL 1
CARE ESTE STRUCTURA CIP-ului?

Cum rezolva CIP-ul o problema?

Daca, de exemplu, trebuie facut urmatorul calcul:

D = A - (B - C)

Prin operatii de "intrare" se introduc:

-programul de prelucrare, prin care automatul programabil este instruit cum sa rezolve problema;

-datele problemei, adica valori pentru A, B si C.

Ce este programul de prelucrare?

Programul cuprinde o multime de instructiuni (comenzi) scrise intr-un limbaj pe care-l intelege calculatorul si ordinea in care vor fi executate.

Instructiunile programului corespund unor pasi elementari in care se rezolva problema.

In exemplul dat, instructiunile pot fi comentate astfel:

1 - cere / citeste valori pentru B si C
2 - calculeaza (B - C)
3 - cere / citeste valoarea lui A
4 - calculeaza D = A - (B - C)
5 - afiseaza (pe un ecran)/scrie (pe hirtie) valoarea lui D
6 -mergi la punctul 1 (pentru a repeta secventa cu alte valori pentru A, B si C)

CIP-ul rezolva problema conform programului introdus si rezultatele prelucrarii facute in circuitele electronice, sint comunicate prin operatii de iesire.

Cum se solutioneaza o problema pe CIP?

Pentru realizarea unei aplicatii pe calculator, se parcurg urmatoarele etape:

-formularea problemei
-stabilirea intrarilor (date furnizate CIP-ului) si iesirilor (ce trebuie sa furnizeze calculatorul?)
-stabilirea modului de rezolvare
-verificarea programului pe calculator
-executarea curenta a programului.

Ce se intelege prin: HARDWARE si SOFTWARE

Specialistii denumesc echipamentele ce alcatuiesc automatul programabil printr-un singur cuvint: HARDWARE, iar programele sint denumite: SOFTWARE.

Schema de functionare a componentelor hardware ale unui automat programabil este urmatoarea (fig. 1.1):

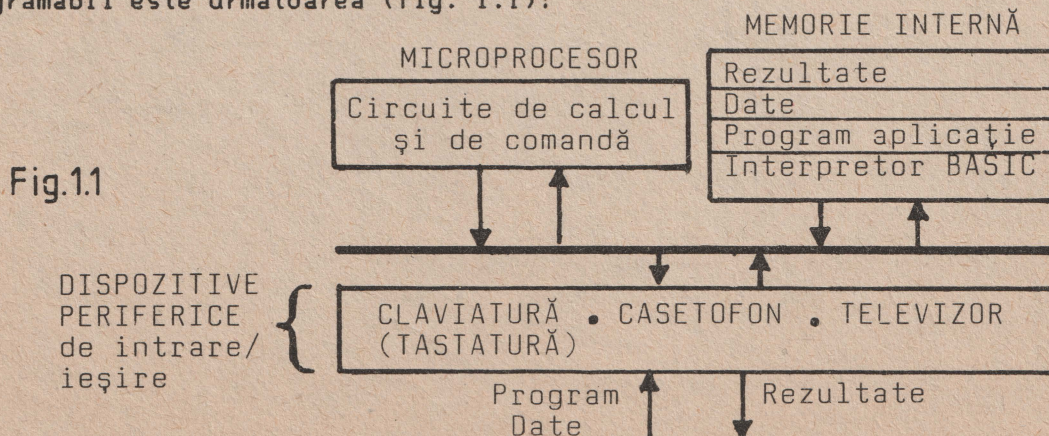


Fig.1.1

Citeva notiuni generale despre fiecare din componente si despre functionarea lor in ansamblu sint prezentate in cele ce urmeaza.

Microprocesorul

Microprocesorul cuprinde pe o mica pastila de siliciu (in engleza CHIP), piese electronice in miniatura care descifreaza, verifica si executa comenzile si calculele prevazute in program.

Nu va nelinistiti! Notiunile au inceput sa fie putin mai complicate, dar ele vor fi reluate si clarificate in capitolele urmatoare.

Memoria interna

In memoria interna, atunci cind lucrati, se afla un interpretor care transforma programul aplicatiei scris intr-un limbaj usor de invatat - limbajul BASIC - intr-un limbaj intern mult mai complicat, specific fiecarui tip de calculator (limbaj in "COD MASINA") care va putea fi executat de microprocesor. Interpretorul permite deci introducerea in memorie (fig.1.1) a diverselor programe de aplicatii (SOFTWARE aplicativ) si executarea lor.

Programele solicita datele necesare si acestea sint memorate intr-o alta zona a memoriei interne. In acest fel, CIP-ul stie ce are de facut si cu ce sa lucreze. Microprocesorul decodifica comenzile si instructiunile si le executa folosind datele disponibile.

Rezultatele prelucrarii sint (fig.1.1) expediate la dispozitivele periferice de iesire.

Configuratia HARDWARE a CIP-ului

Schema functionala descrisa se regaseste concretizata in configuratia hardware a calculatorului.

Configuratia minima a CIP-ului cuprinde:

-microprocesor (tip Z80)
-memorie interna
-circuite de control al afisarii
-circuite de interfata
-difuzor
-tastatura (claviatura)

Se adauga urmatoarele "periferice" de intrare / iesire :

- casetofon
- televizor sau monitor video.

Ce ar fi bine daca ati avea?

O configuratie mai puternica poate include ca dispozitive periferice, imprimanta, joystick si unitate cu disc flexibil. Amplasarea fizica a acestor componente este cea din figura 1.2, iar schema hardware de detaliu, incluzind si alte echipamente ce pot fi legate la CIP, este cea din figura 1.3.

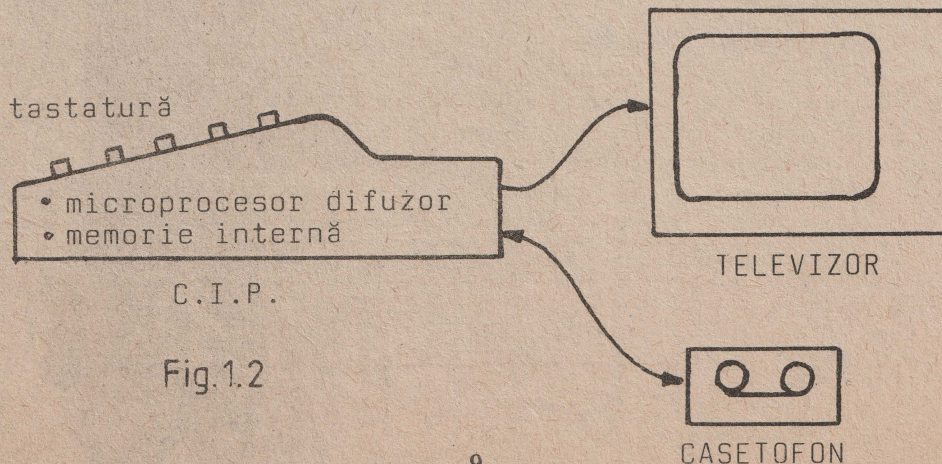


Fig.1.2

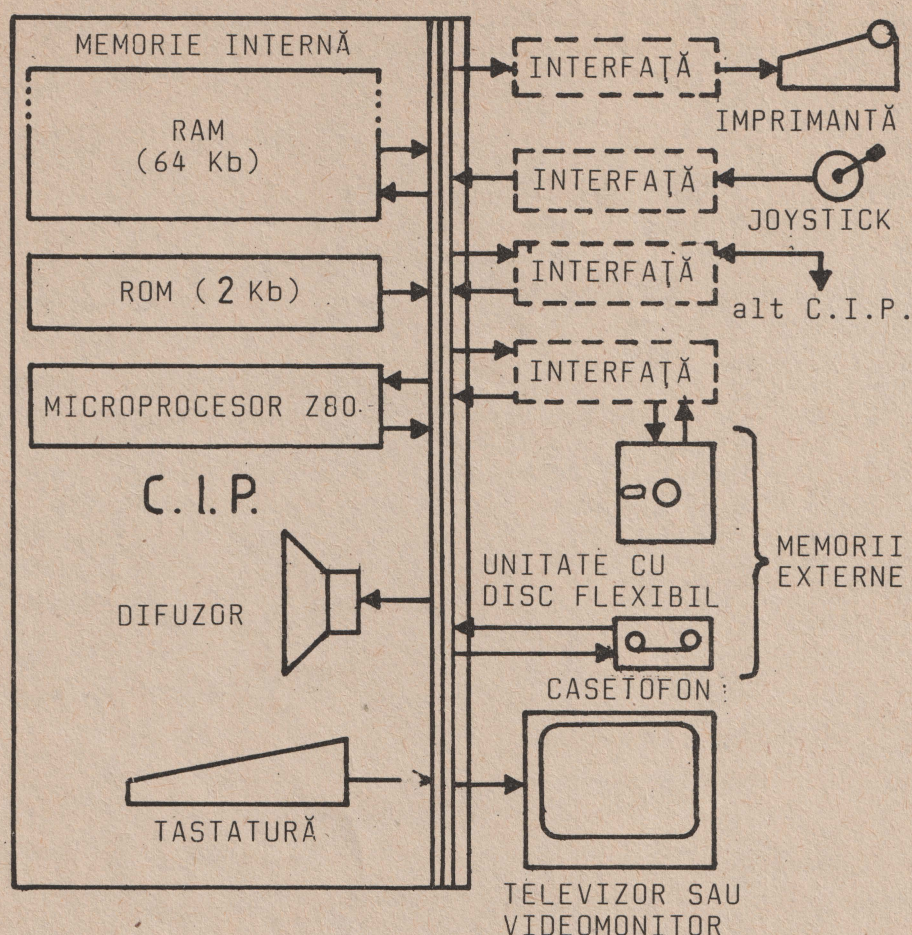


Fig.1.3

Atentie!!!

Poate schema de mai sus vi se pare complicata la prima vedere. Unele din informatiile cuprinse in ea vor fi explicate in capitolele urmatoare, asa incit acum va vom da notiunile generale necesare pentru a incepe sa lucrati cu automatul programabil.

Iarasi despre memoria interna!

Asa dupa cum probabil ati retinut, in memoria interna se introduc informatii prin operatii de "incarcare" sau de "scriere", iar din memorie se "citesc" sau se "extrag" informatiile introduse.

CIP-ul dvs. utilizeaza doua tipuri de memorii:

Memoria ROM

ROM (Read-Only-Memory) - este o memorie al carei continut este stabilit prin fabricatie si care nu poate fi schimbat, ci poate fi numai "citit". Veti constata ca, dupa intreruperea alimentarii electrice a CIP-ului, la repunerea in functiune, programul inscris in ROM va va sta la dispozitie fara a fi nevoie sa-l introduceti dvs. O astfel de memorie se spune ca este "nevolatila". pentru ca nu isi pierde continutul la intreruperea tensiunii de alimentare.

Capitolul 1

Memoria RAM

RAM (Random-Acces-Memory) - este memoria in care se pastreaza programele si datele introduse de utilizator precum si informatii interne necesare functionarii corecte a CIP-ului.

Atentie!!!

Este o memorie "volatila", astfel incit retineti ca la intreruperea alimentarii electrice programele si datele dvs. se pierd (se "volatilizeaza").

Apare o intrebare:

Si daca se intrerupe curentul in mod neasteptat cum facem sa nu pierdem ce am muncit poate citeva ore? Exista o solutie - memoria externa - caseta magnetica - despre care vom vorbi mai departe.

Capacitatea de memorare este limitata de dimensiunea memoriei. Cum se exprima ea si cum se lucreaza cu memoria puteti afla in capitolul 12.

Dispozitive periferice

Dispozitivele periferice sint cele care permit introducerea programelor si datelor sau comunicarea informatiilor de iesire pentru utilizator.

Sint denumite "periferice" spre deosebire de microprocesor, care mai este denumit si "unitate centrala" a aparatului.

CIP-ul necesita ca periferice in configuratia minima:

- -tastatura
- -televizor sau monitor video
- -difuzor
- -casetofon

Intr-o configuratie extinsa, prin achizitionarea unor circuite de legatura speciala (interfete), se mai pot atasa:

- -imprimanta
- -unitate cu disc flexibil
- -joystick

Tastatura

Claviatura CIP-ului este asemanatoare cu a masinii de scris, dar veti afla la capitolul 3 ca ea are un mod deosebit de functionare si de utilizare.

In mod curent este denumita TASTATURA formata din TASTE (clape), iar operatia de utilizare a ei se numeste "TASTARE".

Televizorul

Pentru vizualizarea dialogului cu calculatorul si a rezultatelor prelucrarii puteti folosi televizorul dvs. alb-negru sau color.

Difuzorul

Automatul programabil are incorporat un difuzor care va va semnala incarcarea programelor, sau prin care puteti obtine diferite sunete (melodii).

Casetofonul

Caseta magnetica este folosita ca suport de MEMORIE EXTERNA, deoarece pe ea se pot "salva" (inregistra) programele si datele din memoria interna, pentru a preveni "volatizarea" lor la intreruperea tensiunii de alimentare.

Pentru a "incarca" programe de pe caseta si pentru a le "salva" pe caseta, folositi un casetofon obisnuit, dar in buna stare de functionare.

Un casetofon uzat, cu o mecanica defecta, va va produce multe clipe neplacute cind, avind o caseta cu niste jocuri/programe frumoase, nu veti reusi sa le incarcati sau veti obtine mesajul "Tape loading error" indicind o eroare de banda.

O mica indoitura a benzii, insesizabila in cazul casetei cu muzica nu este acceptata de calculator.

In general, va recomandam sa utilizati acelasi casetofon la "salvarea" si la "incarcarea" programelor, pentru a evita incompatibilitati datorate nealinierii capetelor la doua aparate deosebite.

Caseta magnetica prezinta avantajul unui pret redus, dar are si dezavantajul dificultatii de acces la o anumita informatie, deoarece permite numai o cautare secventiala (program dupa program) de la inceputul benzii, fapt care cere timp si... rabdare.

Discul flexibil

Discul flexibil este un suport de memorie externa care inlatura dezavantajul casetei, el permitind accesul direct si imediat la orice informatie stocata.

Unitatea de disc flexibil este insa un dispozitiv mai scump, si deci mai dificil de procurat.

Imprimanta

CIP-ul permite, prin intermediul unei interfete specializate, cuplarea unei imprimante care sa scrie sau sa deseneze pe hirtie rezultatele prelucrarii.

Joystick-ul

Joystick-ul permite deplasarea pe ecran in orice directie, obtinuta mai usor cu ajutorul unei manete, decit prin tastatura cu ajutorul careia puteti obtine numai "sus" / "jos" / "dreapta" / "stinga"/"foc".

Intrebari recapitulative

Raspundeti in scris pe o foaie separata si confruntati apoi raspunsurile dvs. cu cele corecte prezentate in continuare.

I 1.1 Ce informatii trebuie sa introducem in calculator pentru a rezolva o problema / a realiza un joc / a desena pe ecran ?

I 1.2 Ce echipamente hardware trebuie sa aveti la dispozitie pentru a lucra cu CIP-ul ?

I 1.3 Casetofonul nu este permanent necesar. Cind aveti nevoie de el ?

I 1.4 Sa presupunem ca CIP-ul a fost oprit si il porniti dorind sa introduceti un program de HC-85 /TIM-S. Ce trebuie sa incarcati mai intii de pe caseta?

Raspunsuri

R 1.1 Programul aplicatiei si datele necesare.

R 1.2 CIP-ul + televizor + casetofon + o priza multipla + cabluri de legatura.

R 1.3 a)Cind dorim sa incarcam programe "memorate" pe caseta.
b)Cind am introdus prin tastatura programe mai mari pe care este bine sa le "salvam" pe caseta inainte de oprirea alimentarii CIP-ului.

R 1.4. La orice pornire trebuie sa incarcati de pe caseta interpretorul BASIC-S.

CAPITOLUL 2
PUNEREA IN FUNCTIUNE A CIP-ului

Instalarea

Alegeti un loc de lucru care sa va permita un spatiu confortabil de asezare pentru CIP, televizor (la o distanta corespunzatoare) si casetofon.

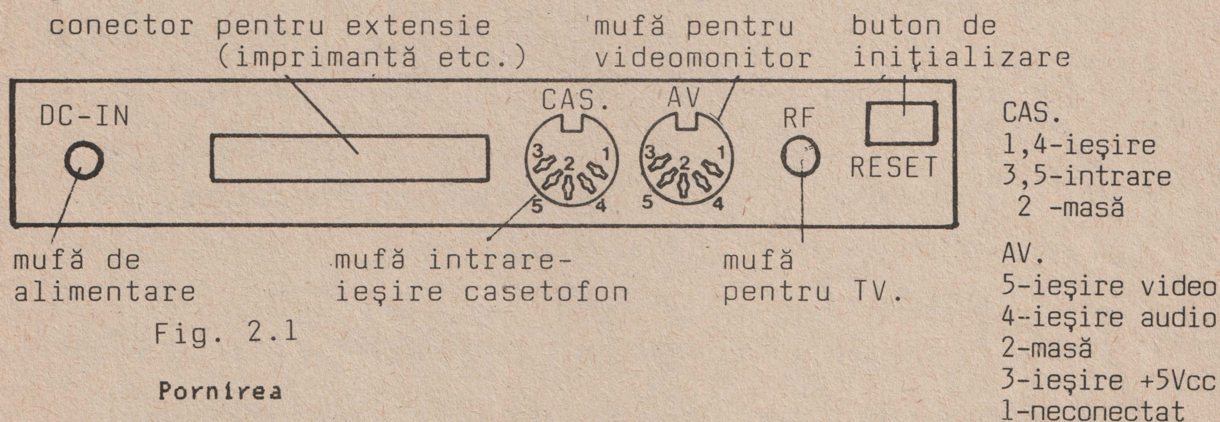
Locul respectiv este recomandabil sa nu fie:

-linga o sursa prea mare de caldura

-linga alte echipamente cu tensiune electrica inalta

Instalarea este extrem de simpla. Va trebuie o priza multipla si...putina atentie.

Priviti din spate CIP-ul si veti observa urmatoarele mufe / conectoare (fig.2.1):



CAS.
1,4-ieşire
3,5-intrare
2 -masă

AV.
5-ieşire video
4-ieşire audio
2-masă
3-ieşire +5Vcc
1-neconectat

Fig. 2.1

Pornirea

Efectuati urmatoarele operatii:

-introduceti fisa alimentatorului in mufa DC-IN a CIP-ului

Atentie!!

Faceti intotdeauna aceasta operatie cu stecher-ul alimentatorului scos din priza, altfel riscati sa se arda siguranta din alimentator.

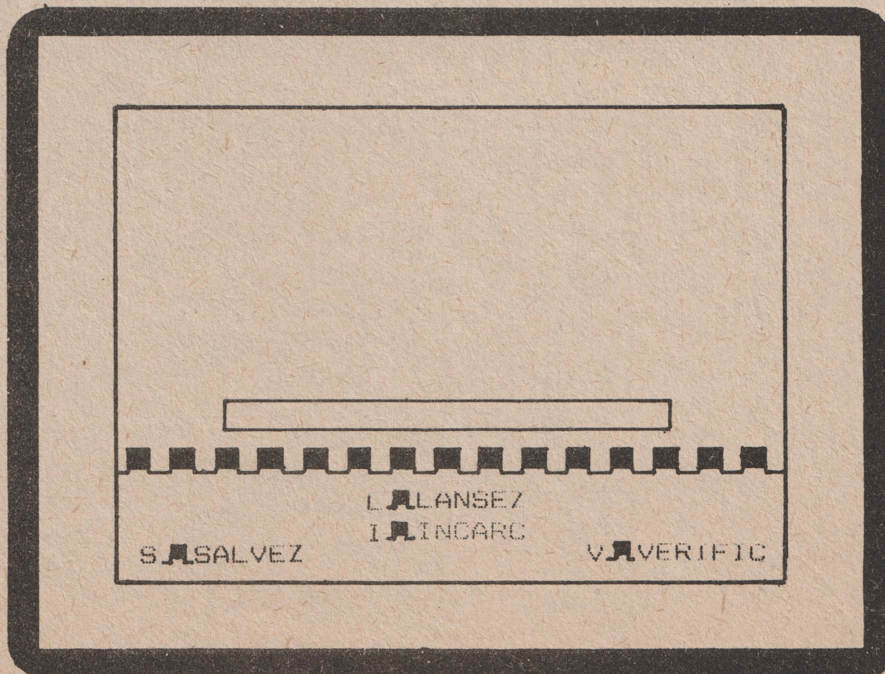
-introduceti cablul de casetofon in mufa notata CAS a CIP-ului.

-introduceti cablul TV in borna de antena a televizorului si in mufa RF a CIP-ului

-porniti TV-ul

-introduceti stecher-ul alimentatorului in priza (220 V)

-acordati TV-ul pe canalele 6-12 pina obtineti o imagine stabila cu urmatorul text:



Initialele din stinga va indica tastele pe care le puteti apasa, pentru a obtine operatia scrisa in dreapta.

Atentie!!

Daca nu apare imaginea descrisa mai sus:

- -verificati daca ati montat corect cablurile si daca mufele sint bine impinse in contacte
- -incercati acordarea TV-ului pe canalul necesar
- -actionati butonul RESET din partea din spate - stinga. Acest buton provoaca initializarea tuturor circuitelor fara a fi oprita alimentarea electrica
- -scoateti din priza si reintroduceti stecher-ul alimentatorului.

Daca nici una din actiunile de mai sus nu au avut ca rezultat obtinerea mesajului corect de inceput, exista ceva defect in configuratia CIP-ului dvs. si veti proceda conform conditiilor de garantie a produsului.

Avind CIP-ul in stare gata de lucru, aveti doua posibilitati:

1 - sa introduceti de pe caseta in memoria CIP-ului un asa numit "INTERPRETOR BASIC" si sa il "LANSATI" in executie, sau un alt interpretor (de ex.PN BASIC O - anexa E);

2 - sa incarcati programe de pe alte casete,in vederea copierii lor (in acest caz cititi anexa D).

Presupunind ca sinteti "incepator", executati aplicatia care urmeaza pentru a invata treptat utilizarea aparatului.

Aplicatie practica AP1

1. Puneti in functiune casetofonul si stabiliti legatura intre el si mufa CAS a CIP-ului.
2. Introduceti caseta demonstrativa in casetofon si pozitionati banda la inceputul fetei a doua.
3. Priviti tastatura si localizati urmatoarele taste:
   - -CAPS SHIFT (CS)
   - -SYMBOL SHIFT (SS)
   - -ENTER (ENT)
   - -SPACE
   - -0 (cifra zero)

In continuare ele vor fi notate in text sub forma prescurtata din paranteze.

Tasta "SPACE" o folositi atunci cind trebuie sa introduceti un "spatiu" sau "blank" (in carte vom folosi pentru spatiu/blank urmatorul desen: "␣" .In exemple, cind veti intilni acet semn, apasati tasta SPACE. Cifra zero apare, atit pe tastatura cit si pe ecran, taiata cu o linie pentru a putea fi deosebita de litera "O".

4. Apasati tasta "L".

In difuzor veti auzi un sunet scurt si pe ecran in dreapta cuvintului LANSEZ va apare un patrat clipitor denumit CURSOR.

5. Apasati tasta ENTER.

Pe ecran clipeste cuvintul CAS (prescurtare de la CASETOFON).

6. Tastati din nou ENTER si porniti casetofonul pentru "citirea" normala a benzii (ca pentru redarea unei melodii).

Cind incepe introducerea programului in memoria interna, se aud in difuzorul CIP-ului niste sunete specifice (cu care va veti obisnui) si pe ecran apar anumite informatii, iar in jur dungi orizontale cu frecventa sunetelor.

Asteptati! Se incarca (aproximativ doua minute) interpretorul BASIC-S.

Informatiile afisate pe ecran va dau amanunte despre programul "citit" in memorie. Semnificatia lor o veti intelege mai tirziu (anexa D).

Daca totul a fost in ordine, pe ecran se afiseaza mesajul BASIC-S aratind ca CIP-ul este pregatit sa execute ce doriti, daca "discutati" cu el in limbajul BASIC.

Atentie!

Daca ati gresit secventa de incarcare sau ceva nu este in ordine (de obicei caseta sau casetofonul) si incarcarea nu decurge cum a fost descrisa mai sus, opriti casetofonul, actionati butonul RESET (din spate stinga), derulati banda la inceput si reluati secventa de la punctul 4.

Avind la dispozitie interpretorul BASIC-S puteti comunica cu CIP-ul prin intermediul tastaturii. Poate la inceput vi se va parea complicat dar prin explicatiile pe care vi le voi da in continuare si prin folosirea casetei demonstrative, va puteti antrena si veti constata ce ingenios este conceputa tastatura.

7. Apasati <ENT> (tasta ENTER)

In coltul stinga-jos al ecranului apare cursorul clipitor, cuprinzind litera K.

CIP-ul asteapta nerabdator ca dvs. sa-i introduceti o comanda in limbajul BASIC-S.

```
8. Introduceti: BORDER   4            <ENT>
                ↑        ↑             ↑
   apasati tasta "B"  tasta 4    tasta ENTER
```

I-ati comandat CIP-ului sa afiseze un contur (BORDER) de culoare verde (culoarea numarul 4). El v-a executat comanda si v-a raportat ca totul a fost in ordine (OK!).

Ati realizat primul dialog cu CIP-ul.

Retineti ca una din principalele caracteristici ale BASIC-ului este acest mod permanent de lucru "conversational" (sau "interactiv"), interpretorul fiind tot timpul gata sa raspunda la mesajele /comenzile /instructiunile dvs. expediate catre el prin apasarea tastei ENTER (o vom nota prescurtat cu <ENT>).

Deci cit timp nu ati actionat aceasta tasta, CIP-ul (interpretorul BASIC) nu stie ce ati introdus dvs. prin tastatura (desi informatiile apar pe ecran) si abia cind actionati <ENT>, el primeste si interpreteaza mesajul dvs. si raspunde corespunzator.

9. Introduceti o alta comanda care va permite sa "afisati" pe ecran litere/cifre/semne, in timp ce exersati utilizarea tastaturii. Afisati alfabetul:

```
PRINT "abcdefghijklmnopqrstuvwxyz"          <ENT>
  ↑   ↑                          ↑            ↑
  |   |_________ghilimele________|            |
apasati |apasati simultan tastele|         apasati
tasta "P"|  SYMBOL SHIFT si P    |          ENTER
```

Daca ati gresit o litera o puteti sterge tastind in acelasi timp CAPS SHIFT si "0" (zero). Se tasteaza 0 (zero) tinind CS apasat. In continuare vom nota aceata apasare simultana cu semnul "+". Deci cind vi se indica (CS+0) dvs.veti apasa in acelasi timp pe CAPS SHIFT si pe tasta 0.

O alta precizare! Limbajul BASIC, asa dupa cum veti constata, este destul de "prietenos" adica permite un dialog destul de usor intre utilizator si CIP, dar are regulile lui stricte conform carora va poate intelege. Daca dvs. nu respectati aceste reguli, CIP-ul refuza sa primeasca mesajul dvs., afisind un semn de intrebare, sau va raspunde cu "VARIABLE NOT FOUND" (variabila negasita). Daca ati ajuns intr-o astfel de situatie, ca neinitiat aveti doua posibilitati:

- stergeti tot rindul apasind simultan CS si 0;
- actionati butonul "RESET" din spate.

Retastati rindul (nu uitati ghilimelele), si expediati mesajul cu ENTER.

Deci:

```
PRINT "abcdefghijklmnopqrstuvwxyz"        <ENT>
```

10. Introduceti alfabetul cu litere mari:

PRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZ" <ENT>

apasati CS si 2
si apoi ABC...

11. Tastati cifrele cu un spatiu intre ele:

PRINT "1␣2␣3␣4␣5␣6␣7␣8␣9␣0" <ENT>

apasati tasta "SPACE"

12. Introduceti o linie cu urmatoarele semne prin tastarile simultane indicate:

PRINT "!@#$%&´()<>;-+=:?/*.," <ENT>

!		(SS+1)
@	(a rond)	(SS+2)
#	(diez)	(SS+3)
$		(SS+4)
%		(SS+5)
&	(AND)	(SS+6)
´	(apostrof)	(SS+7)
(		(SS+8)
)		(SS+9)
<	(mai mic)	(SS+R)
>	(mai mare)	(SS+T)
;		(SS+O)
-	(minus)	(SS+J)
+		(SS+K)
=		(SS+L)
:		(SS+Z)
?		(SS+C)
/	(slash)	(SS+V)
*	(asterisc)	(SS+B)
.	(punct)	(SS+M)
,	(virgula)	(SS+N)

Remarcati ca efectul tastei ENTER este expedierea celor afisate, catre interpretor, dar in acelasi timp si saltul cursorului la inceputul unei noi linii. La alte tipuri de calculatoare aceasta tasta este notata cu "RETURN" sau cu "CR" (CARRIAGE RETURN-retur de car, prin analogie cu intoarcerea la inceputul unei linii la masina de scris).

13. Tastati alaturi urmatoarele litere/cifre/semne pentru a observa diferentele grafice intre ele:

0 (cifra zero) O (litera O) ´(apostrof) ,(virgula) x * (asterisc)

I (litera I) 1 (cifra 1)

Dupa care introduceti comanda:

PRINT "000OOO´´´,,,xxx***III111" <ENT>

apasati CS+I

apasati CS+2

14. Mentinind apasata orice tasta, litera/cifra /semnul respectiv se repeta pina in momentul cind incetam apasarea pe tasta.

Verificati, presupunind ca aveti de subliniat un titlu:

PRINT "Lectia 1" <ENT>

PRINT "========" <ENT>

Pentru a sublinia prin repetarea semnului egal (SS+L), nu este nevoie sa apasati de mai multe ori cele doua taste, ci este suficient sa le mentineti apasate pina ajungeti sub cifra 1 (daca ati depasit-o stergeti cu CS+0 inainte de a apasa <ENT>.

15. Apasati tasta "V" obtinind:

CLS <ENT>

Observati (poate cu surprindere) ca se sterge tot ecranul. Ati introdus o comanda de "stergere" a ecranului pe care interpretorul BASIC a inteles-o si a executat-o prompt (CLS-CLear Screen).

16. Numerotati liniile ecranului astfel:

```
                    apostrof
   ┌───────────────────┼─────────────────┬──────────────┐
   ▼                   ▼                 ▼              ▼
PRINT1'2'3'4'5'6'7'8'9'10'11'12'13'14'15'16'17'18'19'20'21'22
```

17. Priviti mai atent la o tasta, de exemplu la tasta "H". Deasupra ei sint scrise patru caractere ce pot fi obtinute cu ajutorul acestei taste:

GOSUB CIRCLE SQR ↑

Incercati obtinerea lor apasind tastele indicate mai jos, observind litera afisata in cursorul clipitor

(Atentie! dupa afisarea fiecarui caracter, stergeti cu CS+0 pentru a obtine din nou cursorul K).

caracter	tastari	cursor
GOSUB	H	K▶L
CIRCLE	1. CS+SS	K▶E
	2. SS+H	E▶L
SQR	1. CS+SS	K▶E
	2. H	E▶L
↑	SS+H	L

Pentru a obtine "CIRCLE" este nevoie de doua tastari succesive:

1- CS+SS -CS si SS apasate simultan

2- SS+H -SS apasata simultan cu H

Tastatura se afla in orice moment intr-un anumit MOD CURENT de lucru, indicat printr-o litera in cursorul clipitor:

K (KEY WORD-cuvint cheie) -pentru comenzi/instructiuni
L (LETTERS-litere) -pentru litere mici si cifre
C (CAPITALS-majuscule) -pentru litere mari
E (EXTENDED-mod extins) -pentru caractere / comenzi
G (GRAPHICS-mod grafic) -pentru simboluri grafice.

Cum obtineti litere mari/mici?

18. Tastati urmatoarele caractere observind pe ecran litera din cursor si caracterele afisate:

```
cursor          [K]        [L]
                 │          │
                 ▼          ▼
caracter        PRINT       "
                 ▲          ▲
                apasati    apasati
                tasta P     SS+P
```

Apasati simultan (CS+2) si cursorul devine [C] permitind introducerea literelor mari:

```
          ┌[C]
          ▼
PRINT "ABC"
          ▲ (CS+2)
```

Reapasati simultan (CS+2) si cursorul redevine [L],putind introduce litere mici:

```
               [C] [L] [C]
                ↓   ↓   ↓
10 PRINT       ABC abc DEF     <ENT>
                    ↑   ↑
              (CS+2) (CS+2)
```

Cum obtineti videonormal/videoinvers?

19. Atunci cind doriti sa apara pe ecran informatii care sa iasa in evidenta fata de altele, puteti folosi modul de afisare videoinvers, adica in loc de caractere negre pe fondul alb al ecranului (VIDEONORMAL), sa apara caracterele albe pe fond negru (VIDEOINVERS).

Tastati urmatoarele:

```
20 PRINT "videonormal [videoinvers] videonormal"     <ENT>
                     ↑             ↑
                   (CS+4)        (CS+3)
                (INV VIDEO)   (TR VIDEO)
```

Ati remarcat ca trecerea dintr-un mod de afisare in altul se face tastind (CS+4) sau (CS+3).

Daca la apasarea tastei <ENT> dupa linia 10 sau 20, va apare un semn de intrebare in linie, BASIC-S va semnaleaza ca aveti o greseala.

Greseala consta probabil in faptul ca nu ati introdus " dupa PRINT sau la sfirsitul liniei. Intr-un asemenea caz, stergeti linia inapoi cu DEL (CS+0) si retastati-o atent.

Oricum retineti ca BASIC-S nu accepta informatii neconforme cu regulile lui si va semnaleaza prompt orice eroare prin "?"

Daca ati introdus corect linia 10, pe ecran trebuie sa aveti:

```
10 PRINT "ABCabcDEF"
20 PRINT "videonormal [videoinvers] videonormal"
```

Nu va impacientati ca linia 20 continua dedesubt si ca nu este frumos aliniata cu partea de sus.

BASIC-S nu tine neaparat la frumusete, ci la corectitudine si el se descurca cu linii nealiniate dar corecte.

20. Introduceti:BORDER 1   <ENT>

```
                ↑      ↑
                |      └──apasati tasta "1"
                └─────────apasati tasta "B"
```

Ati introdus o comanda BASIC-S care a fost executata imediat, avind ca rezultat stergerea celor doua linii de pe ecran si aparitia unui chenar (o "bordura") de culoare inchisa (albastra).

Apasati:

```
<ENT>   -reapar liniile 10 si 20
RUN     -se executa instructiunile cuprinse in liniile 10 si 20
↑
apasati "R"
```

Pe ecran apare rezultatul executiei:

```
ABCabcDEF
videonormal [videoinvers] videonormal
```

Textul introdus videoinvers ramine si in rezultatul dat de calculator si in acest mod puteti pune in evidenta anumite informatii mai importante de pe ecran.

Caracterele sint scrise in dreptul tastelor, prin trei culori

corespunzatoare modului in care pot fi obtinute.

Dupa o perioada, lucrind curent, va va fi foarte usor sa introduceti orice caracter, numai prin simpla asociere vizuala intre caracterul dorit, pozitia tastei si culoare.

La inceput de mare ajutor va va fi Anexa B in care sint indicate in ordine alfabetica, toate caracterele si tastarile necesare pentru obtinerea lor.

Pentru utilizarea tabelului din anexa va fac citeva precizari:

-cuvintele cheie reprezentind comenzi sau instructiuni, sint precedate de ":" (priviti in Anexa B)

:BORDER
:GOSUB
:LOAD
:SAVE

Ele se obtin prin apasarea tastei indicate, in modul K:

-la inceputul liniei de program
-dupa numarul de linie
-dupa caracterul THEN
-dupa : (doua puncte)

Alte caractere se obtin astfel:

BIN	-(CS+SS) B	-se apasa simultan CS+SS si apoi B
READ	-(CS+SS) A	
FLASH	-(CS+SS) SS+V	-se apasa simultan (CS+SS) si apoi (SS+V)
MOVE	-(CS+SS) SS+6	
*	-SS+B	-se apasa simultan tastele indicate
<=	-SS+Q	-idem-
>=	-SS+E	-idem-

21. Exersati utilizarea anexei B completind rindurile urmatoare:

caracter	tastatura
ASN	---- ----
BREAK	----
:COPY	--
GRAPHICS	---- ----
:NEXT	--
:RUN	--
:STOP	----
◧	---- ----
+	----
=	----
<>	----
cursor sus	----
cursor stinga	----
cursor dreapta	----

Invatati tastatura in dialog cu CIP-ul

Incepeti antrenamentul pentru lucrul cu tastatura folosind caseta demonstrativa.

Deconectati si apoi conectati calculatorul la retea pentru a reveni sub controlul monitorului rezident. Introduceti in casetofon caseta demonstrativa pozitionata la inceputul fetei 1.Tastati comanda "L" <ENT> <ENT>.

Porniti casetofonul si asteptati!

Dupa aproximativ 3 minute si jumatate pe ecran apare emblema CIP -ului.

Asteptati apoi inca 3-4 minute pina cind CIP-ul se prezinta politicos si va invita sa ii dati numele dvs.

Cursorul [L] va face cu ochiul si asteapta.

Introduceti numele si tastati <ENT>.

In continuare urmati indicatiile date de CIP care va va face doua

demonstratii de posibilitatile grafice si...muzicale pe care le are.
Cind v-ati plictisit de desenat poligoane si spirale, la intrebarea:

continuam? [d/n] raspundeti cu "n"!

Dupa ce se va mai lauda cu citeva figuri, aparatul va trece la o munca serioasa si va prezinta o lista "MENIU" de optiuni pentru invatarea lucrului cu tastatura.
Alegeti partea care doriti sa o invatati tastind o cifra intre 1-9.

Intrebari recapitulative

Raspundeti in scris pe o foaie separata si confruntati apoi raspunsurile dvs. cu cele corecte prezentate in continuare:

I 2.1 Ce se afiseaza pe ecran dupa incarcarea interpretorului BASIC, aratind ca totul functioneaza bine si ca putem incepe lucrul?
I 2.2 Ce taste veti folosi pentru:
a) a introduce un spatiu
b) a sterge inapoi
c) a introduce semnele speciale (! ? + - ;)
d) a sterge tot ecranul
I 2.3 Care este efectul actionarii:
a) butonului RESET
b) tastei ENTER
I 2.4 Cum opriti CIP-ul?
I 2.5 Ce trebuie sa introduceti la inceput in CIP, pentru a putea utiliza programe de HC/TIM-S/SPECTRUM?
I 2.6 Daca s-a intrerupt alimentarea electrica, trebuie sa reluati operatia ceruta de intrebarea I 2.5 (da/nu)?

Raspunsuri

R 2.1 BASIC-S
R 2.2 a) SPACE
b) (CS+0)
c) (SS+1) (SS+C) (SS+K) (SS+J) (SS+O)
d) CLS
R 2.3 a) Provoaca initializarea tuturor circuitelor fara a fi oprita alimentarea electrica
b) Expediaza textul afisat catre interpretorul BASIC si muta cursorul la inceputul liniei urmatoare
R 2.4 Intrerupind alimentarea electrica
R 2.5 Interpretorul BASIC-S
R 2.6 Da!

CAPITOLUL 3
CUM COMUNICAM CU CIP-ul?

Din capitolul 1 ati retinut ca un calculator electronic stie sa vae indeplineasca o dorinta, in masura in care l-ati invatat cum sa o faca. Pentru a-l instrui, trebuie sa-i introduceti in memorie o succesiune logica de INSTRUCTIUNI / COMENZI formind un PROGRAM.

Pentru ca sa inteleaga programul si sa execute instructiunile / comenzile, acestea trebuiesc exprimate intr-un limbaj pe care-l "stie" calculatorul .

CIP-ul este astfel construit incit sa inteleaga limbajul BASIC.

BASIC-ul este un limbaj de programare cu o utilizare foarte larga datorita usurintei cu care poate fi invatat si multiplelor lui posibilitati de utilizare.

De ce se cheama BASIC

Numele limbajului BASIC provine din initialele cuvintelor in limba engleza:

Beginner's All purpose Symbolic Instruction Code,

care in traducere libera arata ca este destinat tuturor aplicatiilor si este pentru incepatori.

Intr-adevar, acest limbaj este mai usor de invatat decit limba engleza, el avind un vocabular simplu, cu putine cuvinte, totusi cu reguli foarte stricte.

Chiar daca nu cunoasteti limba engleza, nu va va fi greu sa invatati vocabularul BASIC-ului deoarece cuvintele le veti tot repeta si mai ales, deoarece CIP-ul se va dovedi un "incapatinat", refuzind sa treaca peste greselile dvs. si nu va executa decit ceea ce ii introduceti corect.

Un program BASIC este constituit dintr-o multime de instructiuni scrise in acest limbaj si introduse in CIP pentru a fi executate.

Cum executa CIP-ul un program?

Programul conceput si introdus de dvs. este de obicei numit PROGRAM SURSA. El este tradus de INTERPRETORUL BASIC si transformat intr-un limbaj intern al calculatorului denumit limbaj COD MASINA.

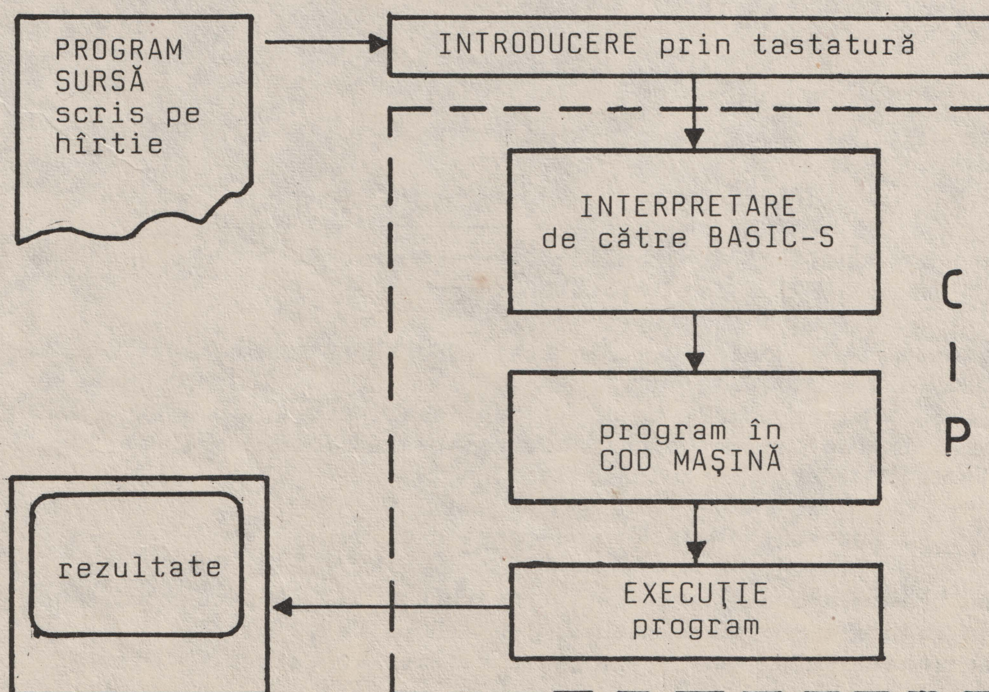


Fig. 3.1

Cum codifica CIP-ul comenzile si instructiunile

Instructiunile / comenzile / datele, pentru a putea fi prelucrate in circuitele electronice ale calculatorului, sint reprezentate sub forma de CARACTERE intr-un cod in care sint numerotate de la 0 la 255.

Priviti in anexa A a manualului si veti gasi in coloana din stinga un numar zecimal reprezentind codul si imediat in coloana din dreapta "caracterul" respectiv. De exemplu:

COD	CARACTER	
13	ENTER	(comanda ENTER)
32	spatiu	(spatiu sau blank)
43	+	(semnul +)
53	5	(cifra 5)
65	A	(litera A)
97	a	(litera a)
245	PRINT	(instructiunea PRINT)
255	COPY	(instructiunea COPY)

Dupa cum observati, CARACTERELE pot fi:

- semne speciale (aritmetice/de punctuatie)
- litere mari
- litere mici
- caractere grafice (▌▯▛ ...)
- instructiuni (PRINT, READ,...)
- comenzi (RUN, LOAD,...)

De exemplu, cind vom introduce urmatoarea instructiune, continutul ei va fi codificat astfel:

```
caractere: PRINT ␣  a  +  b
             ↓   ↓  ↓  ↓  ↓
coduri:     245 32 97 43 98
```

Ca incepator, dvs. nu aveti decit sa concepeti, sa scrieti si sa introduceti programul BASIC-sursa, restul fiind grija CIP-ului.

Mai tirziu, dupa initierea in BASIC, daca veti dori, puteti invata sa lucrati direct in COD MASINA.

Cum arata un program BASIC?

Deocamdata, sa vedem cum arata un mic program scris in limbajul BASIC:

```
10 BORDER 2               <------|
20 INPUT "a=?";a          <------| linii de
30 INPUT "b=?";b          <------| program
40 PRINT "s=";a+b:STOP    <------|
```

Programul este constituit din LINII DE PROGRAM numerotate (linia nr.10, linia nr.20, s.a.m.d.)

BASIC-S admite numere de linii cuprinse intre 1 si 9999. Sint admise numai numere intregi si de obicei, numerotarea se face din 10 in 10, pentru a putea intercala ulterior eventuale linii noi.

Cum se incheie introducerea unei linii?

Avind calculatorul pornit si interpretorul BASIC-S incarcat de pe caseta, introduceti programul de mai sus. Dupa ce ati tastat continutul unei linii, actionati ENTER pentru ca linia sa fie memorata si pentru a se produce saltul la o linie noua.

In timpul introducerii, linia este afisata la baza ecranului.

Linie de program memorata

La actionarea tastei <ENT>, interpretorul BASIC-S verifica daca linia a fost corect introdusa si daca o gaseste astfel, permite memorarea ei, semnalata pe ecran prin afisarea in partea superioara a ecranului.

Se memoreaza o linie cu erori?

Daca dupa <ENT> apare in linie un semn de intrebare, el semnaleaza o eroare. Intr-o astfel de situatie, stergeti cu (CS+O) linia pina la capat si retastati-o corect.

Dupa introducerea liniei 40, pe ecran veti avea programul afisat ca in Fig.3.2.:

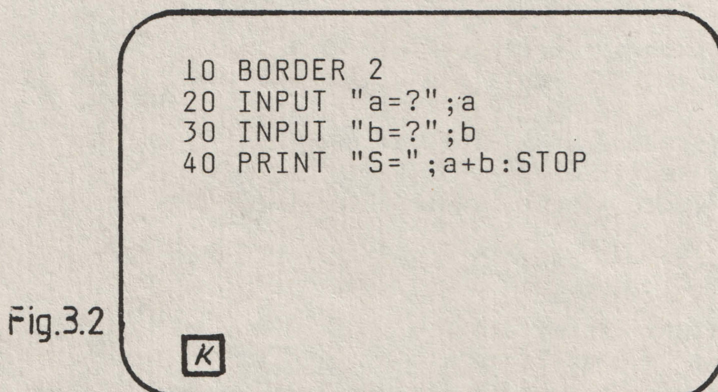


Fig.3.2

Linie de program cu executie imediata

In acest moment programul este memorat si asteapta o comanda pentru a fi executat. Introduceti comanda RUN actionind tasta "R". Ea va fi afisata in dreptul cursorului clipitor [K] si dupa ENTER va declansa executia programului. Retineti ca o linie nenumerotata (de ex. RUN) este executata imediat fara a fi memorata.

Pentru a anticipa ce se va intimpla prin executia programului, sa il comentam linie cu linie:

```
10 BORDER  2            - afiseaza un chenar rosu al ecranului;
20 INPUT"a=?";a  }      - cere valori pentru a si b si le
30 INPUT"b=?";b  }        memoreaza;
40 PRINT"s=";a+b:STOP   - afiseaza suma  s=a+b  si opreste
                          executia programului
```

Cum i-ati introdus comanda RUN, CIP-ul prompt a trecut la treaba. El executa prima linie (coloreaza chenarul) si va cere o valoare pentru a.

Introduceti de exemplu:

a=?10 <ENT>

Acum va cere o valoare pentru b:

b=?20 <ENT>

Aproape instantaneu va afiseaza rezultatul executiei intregului program. (Fig.3.3)

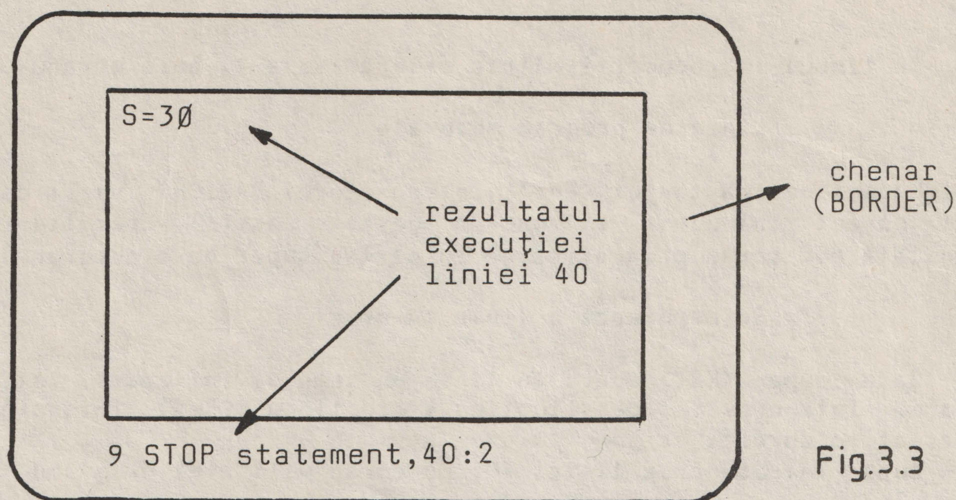


Fig.3.3

Linii de program: -simple
-multiple

Sa analizam putin formatul unei linii de program.
Linia de program poate fi:
-simpla - cu o singura instructiune

```
10 BORDER 2      <ENT>
```

numar linie — instructiune — terminator de linie

-multipla - cu mai multe instructiuni separate intre ele prin ":" (doua puncte)

```
40 PRINT "s=";a+b:STOP      <ENT>
```

o instructiune — alta instructiune

Programul de mai sus poate fi deci introdus pe o singura linie multipla.
Verificati tastind:

```
10 BORDER 2:INPUT"a=?";a:INPUT "b=?";b:
PRINT "s=";a+b:STOP    <ENT>
```

Remarcati ca vechea linie 10 este inlocuita cu noua linie multipla.
Pentru a executa programul, sa stergem liniile 20, 30, 40 devenite inutile:

Cum stergeti o linie?

Stergerea unei linii complete se obtine foarte usor prin introducerea numarului ei urmat de <ENT>.

```
20 <ENT>   - linia 20 dispare
30 <ENT>   - linia 30 dispare
40 <ENT>   - linia 40 dispare
```

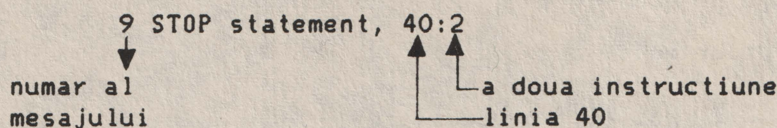
Dati din nou comanda de executie a programului:

```
RUN   <ENT>
a=?10 <ENT>
b=?20 <ENT>
```

Rezultat:

```
S=30
9 STOP statement, 10:5
```

Rezultatul calcului este acelasi, dar difera mesajul de oprire a programului. La prima executie, oprirea s-a facut prin instructiunea STOP, a doua instructiune din linia 40.



La a doua executie oprirea s-a facut prin instructiunea STOP, a cincea in linia 10:

9 STOP statement, 10:5

linia 10 ◄── ──► a cincea instructiune

Inainte de a trece la aplicatia practica, sa recapitulam ce ati invatat din limbajul BASIC pina acum:

-informatiile sint considerate in calculator ca fiind formate din caractere;

- codul conform caruia este construit CIP-ul, cuprinde 255 caractere (vezi anexa A);

- un program se introduce prin linii numerotate (care sint executate numai dupa o comanda speciala) si nenumerotate (care sint executate imediat);

- pe o linie multipla se pot introduce mai multe instructiuni/comenzi, separate prin ":"

- executia unui program se obtine prin comanda RUN;

- modificarea unei linii se obtine prin introducerea noului continut cu acelasi numar de linie;

- stergerea unei linii se obtine prin introducerea numai a numarului ei (linie vida).

Aplicatie practica AP3

1.Anulati programul anterior, introducind comanda imediata:

NEW <ENT> (NEW ← tasta "A")

2.Tastati:

```
10 LET a=7       <ENT>
20 LET b=13      <ENT>
30 LET c=a+b     <ENT>
40 PRINT c       <ENT>
50 STOP          <ENT>
```

Daca ati gresit la introducere, linia nu este admisa cind actionati <ENT> si eroarea va este semnalata cu "?" clipitor.

Sa simulam acum o astfel de eroare pentru a vedea cum procedati.

Tastati linia 30 astfel:

```
30 LET c+a+b     <ENT>
```

(↑ in loc de "=" ati introdus "+", tastele fiind alaturate. Eroarea va este semnalata astfel:)

30 LET c [?] +a+b [L]

Stergeti linia cu DEL (CS+0) si reintroduceti continutul corect.

3. Introduceti comanda de stergere a ecranului (vezi capitolul 2)

>_______ <ENT>

4. Ecranul fiind sters, sa afisam programul aflat in memorie. Introduceti comanda de listare a lui:

LIST <ENT>

Pe ecran reapare programul.

5. Instructiunile LET pot fi traduse astfel:

```
LET a=7          (fie a=7)
LET c= a+b       (fie c=a+b)
```

Instructiunea PRINT afiseaza valoarea lui c.
Incercati sa scrieti ce va fi afisat dupa executia programului:

----

9 STOP statement, _____:__

6. Introduceti comanda de executie a programului si verificati/corectati raspunsul dvs:

-------- <ENT>

Rezultat:

----

9 STOP statement, _____:____

numarul liniei cu instructiunea STOP ↑ ↑ a cita instructiune pe linie

7. Tastati:

10 LET a=127 <ENT>

Observati ca vechea linie 10 este inlocuita cu noul continut introdus.

8. Ce rezultat veti obtine dupa executia programului?

----

9. Executati programul si comparati rezultatul obtinut cu cel intuit de dvs.

----

10. Tastati:

50 <ENT>

11. Comparati programul afisat acum cu cel anterior. Ce observati?

12. Executati programul si analizati ce se afiseaza:

```
140
0 OK, 40:1
|
└─► numarul mesajului
```

Observati ca CIP-ul este un aparat "descurcaret". Chiar fara sa-i spuneti unde sa se opreasca (fara linia 50 - STOP), el a executat programul si v-a dat un mesaj ca totul a decurs bine (OK) oprindu-se singur la prima instructiune a ultimei linii (40:1).

13. Introduceti comanda:

NEW <ENT>

Transcrieti mesajul obtinut:

----

14. Incercati sa afisati programul:

```
LIST    <ENT>
```

15. Incercati sa executati programul;

```
RUN     <ENT>
```

Nu mai puteti obtine nimic pentru ca programul a fost sters din memorie, ca urmare a comenzii NEW, in vederea introducerii unui nou program.

16. Reintroduceti programul:

```
10 LET a=7     <ENT>
20 LET b=13    <ENT>
30 LET c=a+b   <ENT>
40 PRINT c     <ENT>
50 STOP        <ENT>
```

17. Tastati:

```
35 LET d=b-a   <ENT>
45 PRINT d     <ENT>
```

18. Priviti programul afisat si veti constata ca "ascultatorul" CIP a memorat cuminte noile instructiuni la locul potrivit: linia 35 intre 30 si 40 si linia 45 intre 40 si 50. Vom spune ca liniile 35 si 45 au fost "INSERATE" (intercalate) in program.

Mai puteti remarca ceva nou in linia 45:

```
45 > PRINT  d
```



prompter de linie curenta

Semnul ">" reprezinta un prompter (indicator) de LINIE CURENTA si el arata intotdeauna ultima linie introdusa sau corectata.

Linia curenta poate fi modificata prin editarea (afisarea) ei la baza ecranului, printr-o comanda de editare tastind CS+1.

Incercati chiar acum urmatoarea secventa de operatii:

BORDER 5	<ENT>	- apare chenarul, dar dispare programul de pe ecran
	<ENT>	- programul este reafisat
(CS+1)		- la baza este "editata" linia 45
	<ENT>	- linia 45 este repusa la locul ei in program
(CS+7)		- prompter-ul de linie este deplasat in linia 40
(CS+7)		- prompter-ul de linie este deplasat in linia 35
(CS+1)		- este editata ca linie curenta linia 35
	<ENT>	- linia 35 (care poate fi modificata) este repusa la locul ei in program

Prompter-ul de linie poate fi deplasat in sus (CS+7), sau in jos (CS+6), avind astfel posibilitatea sa stabiliti ca linie curenta, orice linie a programului.

Verificati deplasarea prompter-ului de linie tastind (CS+7) sau (CS+6).

19. Analizati programul afisat si incercati sa intuiti care vor fi rezultatele executiei lui:

__________ valoarea lui c

______ valoarea lui d

20. Introduceti comanda de executie si comparati rezultatele.
21. Executati urmatoarea secventa:

```
<ENT>   - programul este reafisat
 RUN    - programul este sters de pe ecran si apar rezultatele.
```

22. Reintroduceti linia 30 cu cite un spatiu /blank intre caractere astfel:

```
30 LET c␣=␣a␣+␣b      <ENT>
        ↑ ↑ ↑ ↑
        spatii
```

Executati programul! Observati ca totul decurge normal. BASIC-ul ignora deci spatiile.

23. Stergeti din program liniile 45 si 35 introducind doua linii vide.
24. Modificati liniile 10 si 20 retastindu-le astfel:

```
10 INPUT "a=?";a      <ENT>
20 INPUT "b=?";b      <ENT>
```

25. Modificati linia 40 prin editare astfel:
    -stabiliti linia curenta dind comanda

```
LIST 40  <ENT>
```

Apasind a doua oara pe <ENT> reapare programul cu prompter-ul in linia 40
- -coboriti linia curenta in pozitia de editare cu (CS+1).
- -deplasati cursorul spre dreapta cu (CS+8), sau spre stinga cu (CS+5) si introduceti ghilimele astfel incit linia sa arate astfel:

```
40 PRINT "c=a+b=";c    <ENT>
```

26. Noul program instruieste CIP-ul astfel:

```
10 INPUT "a=?";a  }        -cere  valorile lui a si b si le
20 INPUT "b=?";b  }         memoreaza
30 LET  c=a+b              -calculeaza c=a+b
40 PRINT  "c=a+b=";c       -afiseaza c=a+b
50 STOP                    -opreste executia programului
```

27. Executati programul:

```
RUN        <ENT>
a=?10      <ENT>
b=?20      <ENT>
c=a+b=30
9 STOP statement, 50:1
```

28. Repetati:

```
RUN       <ENT>     -rezultatul anterior dispare
a=?90     <ENT>
b=?10     <ENT>
c=a+b=100
STOP
```

29. Incercati alta forma de executie a programului:

```
          GOTO 10 <ENT> - rezultatul anterior nu mai dispare de pe ecran
tasta "G"_____↑
          a=?23    <ENT>
          b=?16    <ENT>
```

```
c=a+b=39 <ENT>
STOP
```

Iar:

```
GOTO 10   <ENT>
a=?2530   <ENT>
b=?39870  <ENT>
c=a+b=42400
STOP
```

Observati ca executia programului se poate face cu RUN si in acest caz valorile unei executii anterioare sint anulate, sau cu o instructiune

GO TO n

de trimitere la o linie "n" a programului de unde doriti sa inceapa executia.

In acest ultim caz, valorile anterioare nu sint anulate.

30.Inlocuiti continutul liniei 50 cu urmatorul:

```
50 GO TO 10   <ENT>
```

Acum programul se va executa astfel:

```
┌──►10 (cere o valoare pentru a)
│   ¦
│   20 (cere o valoare pentru b)
│   ¦
│   30 (calculeaza c=a+b)
│   ¦
│   40 (afiseaza c=a+b=valoarea lui c)
│   ¦
└◄──50 (mergi la linia 10)
    RUN
    a=?50      <ENT>
    b=?15      <ENT>
    c=a+b=65   <ENT>
    a=?
```

Deoarece ati inlocuit instructiunea de oprire STOP cu instructiunea GO TO ("mergi la") programul nu s-a mai oprit din executie si va cere in continuare alte si alte valori pentru a si b. Incercati!

31.In momentul in care doriti sa intrerupeti aceasta executie fara sfirsit a programului, la solicitarea unei valori pentru a sau b, introduceti:

```
STOP   <ENT>      (STOP obtineti cu SS+A)
```

Programul va fi oprit cu mesajul:

```
H STOP in INPUT
```

32. Anulati programul existent in memorie si introduceti:

```
NEW
10 PRINT "multe salutari, prietene!"
20 PAUSE 0
30 GO TO 10
```

Executati programul:

```
RUN   <ENT>
```

Veti primi mesajul:

multe salutari, prietene!

Apasati pe orice tasta si veti fi din nou salutat:

multe salutari, prietene!

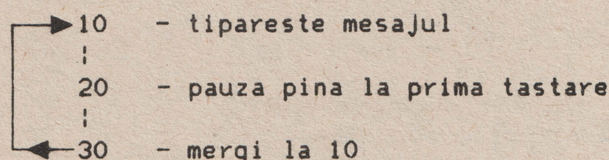
In continuare, la apasarea oricarei taste, veti primi la nesfirsit salutari. Dar pina cind? Pina veti dori sa intrerupeti primirea de mesaje, determinata de fapt, de executia repetata a programului (se spune ca programul "cicleaza", adica a intrat intr-ul ciclu fara sfirsit).

Introduceti comanda de intrerupere BREAK (CS+SPACE) si executia programului va fi intrerupta, afisindu-se mesajul:

L BREAK into program,20:1

Apasati <ENT> pentru a fi reafisat programul si priviti-l cu atentie. Ce s-a intimplat?

Ultima instructiune a programului GO TO 10 a determinat de fiecare data urmatorul ciclu:



Comanda BREAK va permite intreruperea executiei unui program in orice moment, cu exceptia cazului cind CIP-ul asteapta un raspuns la o instructiune INPUT, cind puteti intrerupe executia numai prin STOP, asa cum ati vazut in exemplul precedent.

Executia unui program intrerupta prin BREAK/STOP, poate fi reluata cu instructiunea CONTINUE (prescurtat CONT). Incercati!

O ultima noutate in acest capitol:

Instructiunea PAUSE permite oprirea executiei programului, la dorinta programatorului, astfel:

PAUSE n -opreste executia pe o perioada de (n/50) secunde
PAUSE 0 -opreste executia pina la apasarea oricarei taste.

BREAK <ENT>
_________ <ENT> -stergeti ecranul
_________ -reluati programul intrerupt

multe salutari, prietene!

__________ -intrerupeti programul
__________ -stergeti programul din memorie

Recapitulare: operatii si comenzi pentru elaborare si executie program

-Linia de program curenta (in care se pot face modificari) este indicata de prompter-ul ">". Ea poate fi stabilita prin deplasarea prompter-ului cu ↓ sau ↑

-Modificari intr-o linie de program se fac astfel:

.se pozitioneaza prompter-ul de linie curenta cu (CS+6) sau cu (CS+7).

.se deplaseaza cursorul pe linie cu → (CS+8) sau ← (CS+5), se sterg (CS+0) caracterele nedorite si se introduc cele noi.

.se apasa <ENT>

-Executia unui program:

RUN sau RUN nr. linie
GO TO nr. linie
-Afisarea unui program:
.LIST sau LIST nr. linie
Se afiseaza un ecran incheiat cu
scroll? (alt ecran?)
Puteti raspunde:
.apasind orice tasta, pentru continuarea afisarii;
.tastind "N" pentru a renunta la afisarea programului
-Stergerea unui program din memorie:
.NEW
-Intreruperea executiei unui program:
.BREAK (CS+SPACE)
.STOP (SS+A)
Executia se reia cu: CONT
.PAUSE n -opreste executia pe o perioada de n/50 sec.
.PAUSE 0 -opreste executia programului pina se apasa orice tasta
-Stergerea ecranului
.CLS

Intrebari recapitulative

I 3.1 Cum stergeti o linie dintr-un program BASIC?
I 3.2 Cum inserati o linie noua?
I 3.3 Cum schimbati continutul unei linii deja introdusa?
I 3.4 Care este comanda de afisare a unui program memorat?
I 3.5 Cum stergeti din memorie un program memorat?
I 3.6 Instructiunile fara numar de linie au nevoie de comanda RUN pentru a fi executate?
I 3.7 Cum este indicata linia curenta si cum puteti stabili o linie curenta?
I 3.8 Cum deplasati la stinga si la dreapta cursorul intr-o linie de editare?
I 3.9 Cum puteti intrerupe executia unui program si cum reluati executia?
I 3.10 Cum opriti executia unui program pina la prima tastare?

Raspunsuri

R 3.1 a) Introducind o linie vida cu acelasi numar de linie.
b) Stergind inapoi cu DEL (CS+0) caracter dupa caracter.
R 3.2 Cu un numar de linie intermediar.
R 3.3 Introducind o linie cu acelasi numar, dar cu un nou continut
R 3.4 LIST
R 3.5 Cu ajutorul comenzii NEW
R 3.6 Nu, ele sint imediat executate.
R 3.7 Linia curenta este indicata de prompterul de linie si poate fi stabilita prin deplasarea prompterului cu ↑ sau ↓
R 3.8 Cu → sau ←.
R 3.9 Cu BREAK/STOP; CONT
R 3.10 Cu PAUSE 0

CAPITOLUL 4
CUM CALCULEAZA CIP-UL?
operatii;operatori;constante;variabile;expresii

Deoarece acum ati facut "primii pasi" in BASIC, puteti trece la notiuni de detaliu ale limbajului.

O precizare importanta!

In continuare nu veti mai avea specificata actionarea tastei ENTER. Nu uitati insa ca de fiecare data sa o tastati, pentru ca altfel cursorul va clipi uimit la sfirsitul liniei de program, si informatiile tastate de dvs. vor ramine numai pe ecran, fara a fi cunoscute de calculator.

Aplicatia practica AP4

1.Introduceti urmatorul program:

```
10 INPUT "x=";x:PRINT"x=";x
20 INPUT "y=";y:PRINT"y=";y
30 LET z=x+y
40 PRINT "======"
50 PRINT "z=";z
60 STOP
```

Asa cum probabil deja v-ati dat seama, in linia 30 se face o operatie de adunare intre doua numere, care va sint cerute in liniile 10 si 20.

Executati programul:

```
---      (nu uitati ENTER!)
x=14.7  (In BASIC numerele zecimale se scriu cu punct, nu cu virgula)
y=15.3
======
z=30
```

2.Schimbati linia 30 astfel:

```
30 LET z=x-y
```

Inainte de a executa programul, scrieti mai jos care va fi rezultatul daca veti introduce x=20 si y=6

z=________

Acum executati programul si comparati rezultatul cu cel presupus de dvs.

3.Introduceti:

```
30 LET z=x*y
```

Afisati programul.

Executati programul cu x=8 si y=3

z=______

Ce operatie aritmetica se noteaza cu "*"? _______________

4. Introduceti:

```
30 LET z = x / y
```

Afisati programul si executati-l atribuind variabilelor valorile x=36 si y=4.

z=__

Ce operatie aritmetica se noteaza cu "/"? ____________________

5.Incercati:

```
30 LET z=x+y-y/4
```

Afisati programul si studiati-l putin. Ce rezultat credeti ca se va obtine daca introduceti x=6 si y=12.

z=________

Executati programul si comparati rezultatul cu cel anticipat de dvs.

6. Anulati programul si introduceti:

```
10 LET a=5*5
20 LET b=5↑2            ("↑" se obtine cu SS+H)
30 PRINT "a=";a
40 PRINT "b=";b
50 STOP
```

Executati programul si veti remarca obtinerea aceluiasi rezultat. Prin "↑" in BASIC se noteaza ridicarea la putere sau exponentierea, asa incit in linia 20 se calculeaza "cinci la puterea a doua" ceea ce este echivalent cu linia 10.

7.Schimbati:

```
10 LET a=5*5*5
20 LET b=5↑3
```

Cit credeti ca se va obtine?

a=_____

b=_____

Executati programul si confruntati rezultatele.

8. Completati operatorii aritmetici folositi pentru urmatoarele operatii:

scadere _____

exponentiere ______

inmultire ______

impartire ______

adunare _______

9. Anulati vechiul program si introduceti o instructiune imediata (fara numar de linie):

```
PRINT "a=3+4*8/2=";3+4*8/2
```

Notati rezultatul

a=___________=______

Sa analizam putin ce s-a intimplat:

-instructiunea a fost executata imediat (fara RUN);

-instructiunea PRINT a executat calculul expresiei si apoi a afisat rezultatul;

-operatiile s-au facut intr-o anumita ordine. Instructiunea respectiva este transcrierea in limbajul BASIC a expresiei aritmetice:

$$a = 3 + 4 \times \frac{8}{2}$$

CIP-ul a efectuat mai intii inmultirea si impartirea si apoi a facut adunarea.

10.Introduceti, notati, analizati si comparati rezultatele:

```
PRINT "a=(3+4)*8/2=";(3+4)*8/2      rezultat:____
PRINT "a=3+(4*8)/2=";3+(4*8)/2               ____
PRINT "a=3+4*(8/2)=";3+4*(8/2)               ____
```

Calculatorul a efectuat calcule in urmatoarea ordine:

1.paranteza
2.inmultirea/impartirea
3.adunarea

Daca intilneste paranteze evalueaza prioritar continutul lor.

11.Introduceti si notati rezultatele:

```
PRINT 2*100                    -------
PRINT 2*100*100                -------
PRINT 2*100*100*100            -------
PRINT 2*100*100*100*100        -------
```

Remarcati ca rezultatul ultimei operatii a fost afisat sub forma 2E+8 in loc de 200000000.

12.Stergeti ecranul (CLS) si introduceti:

```
                             rezultat
                             --------
PRINT 2/100                  .02
PRINT 2/(100*100)            -----
PRINT 2/(100*100*100)        -----
```

Ultimul rezultat in loc de 0.000002 a fost afisat sub forma: 2E-6 (2E minus 6) unde "E-6" arata ca este un numar zecimal in care cifra 2 se afla dupa virgula in a sasea pozitie (cu zero-uri in fata).

13.Scrieti in forma acceptata de interpretorul BASIC urmatoarele expresii aritmetice. Cum credeti ca vor fi afisate rezultatele?

10×10^7 -------

$10 : 10^6$ -------

Introduceti instructiunile corespunzatoare si verificati forma de afisare a rezultatelor.

14.Introduceti:

```
10 PRINT 17
20 LET a=17:PRINT a
30 LET a=39:PRINT a
```

Executati programul!

In linia 10 cerem calculatorului sa afiseze numarul 17. In linia 20 cerem acelasi lucru, dar prin intermediul literei "a" careia ii vom putea da si alte valori. Limbajul BASIC va permite sa lucrati atit cu valori CONSTANTE (neschimbate) cit si cu VARIABILE (nume care pot lua diferite valori).

Constantele si variabilele sint admise in anumite conditii despre care veti afla multe in continuarea acestui capitol. Daca nu veti respecta aceste conditii (din necunoastere sau neatentie) interpretorul BASIC va refuza sa le ia in considerare semnalind eroarea cu un semn de intrebare in linia de editare.

15.Incercati validitatea unor constante numerice introducind valorile date pentru x si notind ce se afiseaza:

```
NEW
10 INPUT "x=";x:PRINT x
20 GO TO 10
RUN
```

valori pentru x:	ce se afiseaza:
123	123

```
3.14                            ---------
12345678                        ---------
123456789                       ---------
0.123                           ---------
 .123                           ---------
+210                            ---------
-210                            ---------
12abc                           12 [?] abc     ( eroare! nu sint admise litere )
```

Stergeti literele si valoarea va fi admisa.

16.Schimbati linia 10:

```
x=STOP    <ENT>  (STOP obtineti cu SS+A)
10 INPUT "x=";x$:PRINT x$
RUN
```

valori pentru x:	ce se afiseaza:
abc	-----
12abc	-----
BaSiC	-----
aceasta e o VARIABILA SIR	------------------

Prin adaugarea caracterului "$" variabila x este definita ca o VARIABILA SIR in care se pot introduce orice caractere alfanumerice.

17.Actionati butonul RESET si introduceti cu atentie:

```
10 INPUT "caracter1:␣";a$,"caracter2:␣";b$
20 IF a$>b$ THEN PRINT a$;"␣>␣";b$:GO TO 10
30 IF a$<b$ THEN PRINT a$;"␣<␣";b$:GO TO 10
40 PRINT a$;"␣=␣";b$:GO TO 10
```

Programul compara doua caractere introduse la cerere si afiseaza relatia dintre ele, folosind operatorii BASIC

- \> (mai mare decit...)
- < (mai mic decit...)
- = (egal cu...)

Compararea se face pe baza codurilor ASCII ale caracterelor, indicate in anexa A.Priviti in anexa A:

-litera "F" are codul 70, iar litera "E" are codul 69.
 Rezulta ca: F>E (pentru ca 70>69)
-cifra "7" are codul 55, iar litera "d" are codul 100.
 Rezulta ca: 7<d.
-semnul "%" are codul 37, iar semnul "?" are codul 63.
 Rezulta ca: %<?.

Verificati executind programul si confruntind rezultatele cu codurile din anexa A:

```
caracter1: 4        caracter2: 9
            4<9
caracter1: k        caracter2: A
            k>A
caracter1: ↑        caracter2: +
            ↑>+
caracter1: p        caracter2: B
            p>B
```

18.Introduceti:

```
RESET
10 CLS:INPUT "a=(5...10)=␣";a:PRINT "a=␣";a
20 IF a>=5 AND a<=10 THEN CIRCLE 125, 85, 30
:PAUSE 200: GO TO 10
30 IF a<5 OR a>10 THEN PRINT AT 18,1;FLASH 1;
```

```
"numar in afara intervalului!":PAUSE 300:GO TO 10
```

Programul va cere sa-i introduceti un numar cuprins intre 5 si 10 si in functie de raspunsul dvs. reactioneaza astfel:

1)daca numarul introdus va fi mai mare decit sau egal cu 5 (a >= 5) SI (in engleza AND) mai mic decit sau egal cu 10 (a <= 10), adica daca va fi cuprins in intervalul [5,10], CIP-ul va va oferi ca premiu un cerc in mijlocul ecranului; (cu instructiunea CIRCLE).

2)daca numarul introdus va fi mai mic decit 5, SAU (in engleza OR) mai mare decit 10, CIP va va atrage atentia clipind intunecat ca numarul nu se incadreaza in valorile premiabile (cu instructiunea FLASH 1).

Executati programul si verificati vigilenta calculatorului (de ex. cu a=7, apoi a=3).

Pe scurt, programul poate fi comentat astfel:

-sterge ecranul
-linia 10
  . cere un numar intre 5 si 10
  . afiseaza numarul introdus
-linia 20
  . daca numarul este intre 5 si 10 deseneaza un cerc
  . asteapta 4 secunde si apoi intoarce-te la linia 10
-linia 30
  . daca numarul este mai mic decit 5 sau mai mare decit 10 afiseaza clipitor "numar in afara intervalului "
  . asteapta 6 secunde si apoi intoarce-te la linia 10.

19.Incercati acum daca CIP-ul este suficient de "destept" incit sa inteleaga si informatii spuse invers. In programul urmator sa-i cereti sa se descurce in urmatoarea situatie:

-ii memorati printr-o variabila sir numele unei pasari cunoscute: "cuc";

-el sa va ceara un nume de pasare;

-daca vreti sa-i incercati rabdarea nu-i dati la inceput ce stie el, ci orice alte nume de pasari, caz in care sa va ceara cu insistenta pasarea pe care o tine el minte;

-cind ii veti satisface dorinta, sa afiseze bucuros cintecul cucului!

Introduceti:

```
RESET
10 CLS:LET p$="cuc"
20 INPUT "o pasare:␣";x$
30 IF NOT p$=x$ THEN PRINT "alta pasare!":PAUSE 50: GO TO 10
40 PRINT AT 10,12;FLASH 1;"cucu!!!":PAUSE 200: GO TO 10
```

In program este utilizat operatorul logic NOT (in romaneste "nu") pentru a pune conditia: daca p$ NU este egal cu x$ ATUNCI solicita din nou un sir pentru x$.

Si acum dupa ce ati constatat in dialog direct cu CIP-ul cum calculeaza, daca nu toate notiunile va sint foarte clare, ele vor fi reluate cu explicatii suplimentare si in exercitii/intrebari la sfirsitul fiecarui capitol.

Notiuni de baza. Operatori

Limbajul BASIC foloseste urmatoarele tipuri de operatori:

a) operatori aritmetici:

- \+ adunare
- \- scadere
- * inmultire
- / impartire
- ↑ ridicare la putere (exponentiere)

b) operatori relationali
- = egalitate
- <> neegalitate
- > mai mare decit
- < mai mic decit
- >= mai mare decit sau egal cu
- <= mai mic decit sau egal cu

c) operatori logici
- AND "si"
- OR "sau"
- NOT negatie

d) operator de alipire (concatenare) siruri
- + plus

Prioritati in executie

Prioritatea (ordinea) in executie a operatiilor este urmatoarea:
1- expresiile din paranteze
2- ↑ (ridicarea la putere/exponentierea)
3- *,/ (inmultirea,impartirea)
4- +,- (adunarea,scaderea)
5- =,<,>,<=,>=,<>
6- NOT
7- AND
8- OR

O expresie cu operatori de aceeasi prioritate se executa de la stinga la dreapta.

De exemplu expresia $3 + \frac{4^2}{8} - 2$ se scrie in limbajul BASIC 3+4↑2/8-2 si se executa astfel:

1. 3+16/8-2
2. 3+2-2
3. 5-2

rezultat:3

Incercati:

```
NEW
PRINT 3+4↑2/8-2
```

Exemplu: $[3(4 \times 5 + 6^2) + 7](4^2 - 3)$

Expresia in limbajul BASIC se scrie astfel:

```
(3*(4*5+6↑2)+7)*(4↑2-3)
 |   └──1──┘  |   └3┘
 └─────2──────┘
```

Ordinea in executie este:
1- 6↑2
2- 4*5+36
3- 3*56+7
4- 4↑2
5- 16-3
7- 175*13

Calculati singuri si notati rezultatul: _______

Verificati cu o instructiune PRINT imediata.

Probabil v-ati intrebat deja: dar radicalul sau extragerea de radacina cum se calculeaza?

Pentru exemplificare sa scriem linia de program pentru calculul ipotenuzei unui triunghi dreptunghic. Fie variabilele:

a,b - catetele triunghiului c - ipotenuza

Formula aritmetica de calcul este:

$$c = \sqrt[2]{a^2 + b^2}$$

In BASIC se poate scrie in doua feluri:

(a↑2+b↑2)*0.5 sau SQR(a↑2+b↑2)

Constante numerice

Constantele numerice pot avea:

-format intreg:
16 sau +16
-305

-format zecimal (real):
6.25 (echivalent cu 6,25)
0.95 sau .95 (pentru 0,95)

-format exponential:
E ± n (echivalent cu $10^{\pm n}$) unde n este un numar intreg cuprins intre -38 si +37.

Exemple:

-1234=-1.234E+3=-.1234E4
5000000=5E6=5E+6
-.0000123=-1.23E-5=-123E-7

Introduceti si verificati afisarea aceluiasi rezultat:

```
PRINT -.0000123
PRINT -1.23E-5
PRINT -123E-7
```

Constante sir de caractere

Orice sir de caractere alfanumerice introduse intre "" constituie o constanta sir.

Exemple:

"abc"
"NR.␣CRT"
"Nu opriti banda, se incarca programul!"

Daca aveti nevoie sa folositi in sirul de caractere chiar caracterul " (ghilimele), il introduceti de doua ori pentru fiecare aparitie in sir.

```
10 PRINT "CIP"       -afiseaza: CIP
20 PRINT """CIP"""   -afiseaza: "CIP"
```

Variabile numerice

Variabilele numerice pot fi formate din oricite caractere alfanumerice, cu conditia sa inceapa cu o litera:

Exemple:

a
p237
material58

Introduceti:

```
10 LET lungimea=8:LET latimea=6
20 LET suprafata=lungimea*latimea
30 PRINT"suprafata=";suprafata
RUN
```

In acest program aveti:

-constante numerice: 6 si 8
-constante sir: "suprafata="
-variabile numerice: lungimea, latimea, suprafata

Variabile sir de caractere

Puteti defini in program variabile carora sa le puteti atribui diferite

siruri de caractere alfabetice si numerice.

Numele unei variabile sir trebuie sa contina o singura litera urmata de semnul $.

Exemple:

```
A$="DA sau NU"
x$="SALUT!"
```

Un sir de caractere poate fi utilizat si partial sub forma de SUBSIR.

Manevrarea subsirurilor se face cu

S=(n1 TO n2)

- S este o constanta sir sau o variabila sir
- n1 primul caracter al subsirului
- n2 ultimul caracter al subsirului.

Fie sirul: a$="abcde"

Verificati manevrarea lui introducind urmatoarele instructiuni cu executie imediata:

	Ce se afiseaza
CLS	
LET a$="abcde"	
PRINT 12345	12345
PRINT a$	abcde
PRINT a$(3)	c
PRINT a$ (2 TO 4)	bcd
PRINT a$ (␣TO 3)	abc
PRINT a$ (2 TO␣)	bcde
PRINT a$ (2 TO 2)	b
PRINT a$ (4 TO 3)	linie vida (blank)
PRINT a$ (1 TO 5)	abcde
PRINT a$ (1 TO 6)	mesaj de eroare deoarece sirul are numai 5 caractere

Iata un exemplu mai sugestiv.

Daca introducem intr-un sir lunile anului prescurtate in trei litere astfel: l$="ian feb mar apr mai iun"

Putem extrage pentru un eventual calendar subsiruri. Introduceti si verificati!

```
NEW
10 PRINT "12345678901234567890123456789012"
20 LET l$="ian␣feb␣mar␣apr␣mai␣iun"
30 PRINT l$
40 PRINT "luna 2-a:";l$(4 TO 7)
50 PRINT "trim.1:";l$(␣TO 11)
60 PRINT "primavara incepe in luna";l$(8 TO 11)
70 PRINT "semestrul 1:␣";l$
RUN
```

Notatia exponentiala a numerelor

Numerele foarte mari si cele foarte mici pot fi exprimate sub o forma exponentiala astfel:

1234000000=<u>1.234</u><u>e+9</u>

mantisa — exponent (se poate nota cu "E" sau "e")

Mantisa reprezinta numarul zecimal (in locul virgulei se foloseste punctul), iar exponentul este format din litera "E" urmata de un numar care arata pozitia punctului zecimal.

Daca exponentul este pozitiv, numarul respectiv se obtine prin deplasarea spre dreapta a punctului zecimal.

3.7e12=3700000000000

→12 pozitii

Daca exponentul este negativ, numarul respectiv se obtine prin deplasarea spre stinga a punctului zecimal.

```
56e-8=0,00000056
8 pozitii<----┘
```

Exemplul urmator va va ajuta sa intelegeti mai bine reprezentarea exponentiala:

31400	3.14e4
3140	3.14e3
314	3.14e2
31,4	3.14e1
3,14	3.14
0,314	3.14e-1
0,0314	3.14e-2
0,00314	3.14e-3
0,000314	3.14e-4
0,0000314	3.14e-5

Daca CIP-ul va va afisa un rezultat sub forma exponentiala, dvs. veti putea obtine forma zecimala obisnuita destul de usor, luind in considerare semnul exponentului.Daca este + scrieti punctul zecimal la dreapta cu atitea pozitii cite arata exponentul.

```
9.876543E8 este 987654300
                └------> 8 pozitii
```

Daca semnul este negativ, scrieti punctul zecimal la stinga cu numarul indicat de pozitii.

```
23e-6 este 0,000023
   6 pozitii<----┘
```

Operatori relationali

Un program BASIC se executa "secvential" linie dupa linie, dar aveti posibilitatea sa-i spuneti calculatorului, ca in anumite conditii sa nu mai repecte secventa crescatoare a numerelor de linii. Pentru aceasta veti putea folosi instructiunea de testare conditionata IF...THEN...(in traducere: DACA...ATUNCI...).

IF conditie THEN actiune

Cind conditia este indeplinita se executa actiunea indicata dupa THEN.

Conditia (sau testul) care urmeaza dupa IF o puteti pune folosind operatori relationali.

operator	semnificatie	exemple (vezi anexa A)	
>	mai mare decit	4>3	b>a
<	mai mic decit	3<4	a<b
>=	mai mare decit sau egal cu	x>=y	n>=8
<=	mai mic decit sau egal cu	p<=q	t<=10
<>	nu este egal cu (diferit de)	5<>7	m<>N
=	egal cu	K$="ABC"	x=4

Operatori logici

In instructiunea IF veti putea utiliza operatori logici:

AND (in traducere: SI)
OR (SAU)
NOT (NU)

AND

Folosind AND intre doua conditii, impuneti ca amindoua sa fie indeplinite pentru ca sa se execute ce urmeaza dupa THEN

```
IF conditia1 AND conditia2 THEN...
```

Pentru a intelege mai bine, inchipuiti-va o teava prin care curge apa si care are doua robinete R1 si R2:

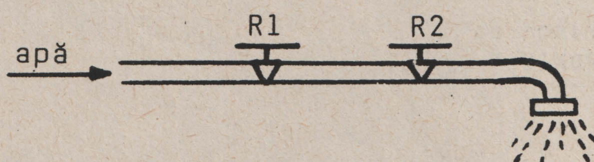


Fig. 4.1

Ele pot fi in urmatoarele combinatii:

R1	R2	Efect
inchis	inchis	nu curge apa
inchis	deschis	nu curge apa
deschis	inchis	nu curge apa
deschis	deschis	apa curge

Conditiile se pun deci astfel:

DACA R1=deschis SI R2=deschis ATUNCI apa curge

In BASIC-S veti scrie:

```
IF R1=......AND R2=......THEN......
```

OR

Operatorul logic OR il veti folosi atunci cind din doua conditii este suficient una sa fie indeplinita, pentru ca sa se execute ce urmeaza dupa THEN.

Inchipuiti-va acum teava de apa ramificata astfel:

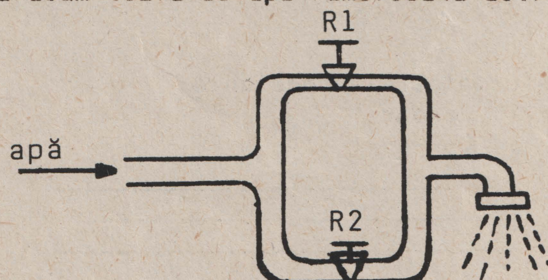


Fig. 4.2

Se observa imediat ca un singur robinet este suficient sa fie deschis pentru ca apa sa curga.

Conditiile se pun astfel:

DACA R1=deschis SAU R2=deschis ATUNCI apa curge

In BASIC-S veti scrie:

```
IF R1=..... OR R2=.....THEN.....
```

NOT

Operatorul NOT se foloseste inaintea unei conditii care daca NU este indeplinita, se executa ce urmeaza dupa THEN.
Exemplu:
DACA NU este x diferit de y ATUNCI scrie x=y
IF NOT conditie THEN......

Alipirea (concatenarea) sirurilor

Puteti reunii doua siruri de caractere intr-unul singur, folosind operatorul de concatenare (alipire)"+"

```
a$=b$+c$
```

Presupunind ca variabilele sir sint :

```
LET b$="Intreprinderea"
LET c$="␣ELECTRONICA"
```

dupa executarea liniei:

```
LET a$=b$+c$:PRINT a$
```

se va afisa pe ecran:

```
Intreprinderea ELECTRONICA
```

Intrebari recapitulative si exercitii

I 4.1 In executia operatiilor aritmetice exista o ordine (prioritate). Scrieti in ordine inversa ce operatii se executa:

4) _________ (ultimele)
3) _________
2) _________
1) _________ (primele)

I 4.2 Scrieti forma BASIC pentru calculul expresiilor:

1) $$d = \frac{a \times b}{c} + 10$$

2) $$e = \frac{(a + 5)\, b^n}{2,7\, (c - \frac{d}{b}) + 1}$$

3) $$y = ax^2 + bx + c$$

I 4.3 Gasiti erorile in urmatoarele constante:

	constanta	eroare
a)	4.320E2.1	_____________________________
b)	1,05	_____________________________
c)	5.666E+42	_____________________________
d)	0.702-E6	_____________________________

I 4.4 Ce rezultat va fi afisat dupa executarea urmatorului program:

```
10 LET A=5:LET B=8:LET C=(A*B+2)/7:PRINT C
```

I 4.5 Cum vor fi afisate urmatoarele numere:

a) 1200000000 _______________
b) 0,000000444 _______________

I 4.6 Care este forma zecimala obisnuita a urmatoarelor constante

numerice:

a) 3.001E+10 _______________

b) 1.989E-6 _______________

I 4.7 Date fiind urmatoarele variabile:

M,M$,Abc,P123,X$,sir,doi,S$

precizati care sint:

a) variabile numerice:___________

b) variabile sir:_______________

I 4.8 Ca urmare a instuctiunii:

```
IF a<=13 AND b>5 THEN PRINT x
```

se afiseaza x daca:

1) a=13;b=13 ___ (raspundeti da/nu)

2) a=20;b=20 ___ (da/nu)

I 4.9 Daca introduceti:

```
10 LET D$="am CIP"
20 IF b$="stiu BASIC" OR D$="am CIP"
THEN PRINT "sint bucuros!"
30 PRINT "regret!"
```

dupa executia programului veti primi raspunsul "sint bucuros!"?

a) ___ (da/nu)

b) De ce? ______________________________________

I 4.10 Atentie o intrebare putin mai dificila!

Judecati cu atentie si incercati sa raspundeti, apoi introduceti programul si verificati-va rationamentul initial.

Fie urmatorul program:

```
 5 PRINT
10 INPUT "x=";x,"y=";y
20 PRINT "x=";x,"y=";y
30 IF NOT x<>y THEN PRINT "AAAA":GOTO 5
40 PRINT "BBBB"
```

Ce mesaj va fi afisat daca introducem:

a) x=4 y=1 ______

b) x=8 y=8 ______

c) x=1 y=4 ______

I 4.11 Fie urmatorul program:

```
 10 PRINT "introduceti pentru y valoarea 1,2 sau 3"
 20 INPUT "y=";y
 30 IF y=1 THEN GOTO 70
 40 IF y=2 THEN GOTO 80
 50 IF y=3 THEN GOTO 90
 60 GOTO 100
 70 PRINT "ROSU":GOTO 20
 80 PRINT "ALB":GOTO 20
 90 PRINT "VERDE":GOTO 20
100 STOP
```

a)Ce se va afisa dupa executie daca:

y=2 __________________________

y=1 __________________________

y=3 __________________________

b)Care sint in program constantele numerice?

c)Care sint variabilele numerice?

d)Care sint constantele sir?

e)Ce se intimpla daca introduceti pentru y o valoare diferita de 1, 2 sau 3?

Raspunsuri

R 4.1 4) adunarea/scaderea
3) inmultirea/impartirea
2) exponentierea
1) expresiile din paranteze

R 4.2 1) d=(a*b)/c+10
2) e=((a+5)*b↑n)/(2.7*(c-d/b)+1)
3) y=a*x↑2+b*x+c

R 4.3 a) exponentul nu este numar intreg
b) virgula nu este admisa
c) exponent prea mare (>37)
d) exponent gresit (trebuie:E-6)

R 4.4 6

R 4.5 a) 1.2E+9
b) 4.44E-7

R 4.6 a) 30010000000
b) 0,000001989

R 4.7 a) M, Abc,P123,sir,doi
b) M$,X$,S$

R 4.8 1) da 2) nu

R 4.9 a) da!
b) pentru ca OR cere sa fie indeplinita o singura conditie.

R 4.10 Linia 30 a programului se citeste asa:
daca NU este x diferit de y (adica x=y) afiseaza "AAAA" si intoarce-te la linia 5.

In cazul in care conditia pusa in linia 30 nu este indeplinita (adica x este diferit de y) programul trece la linia 40 si afiseaza mesajul corespunzator.

Deci raspunsurile corecte sint:
a) BBBB
b) AAAA
c) BBBB

R 4.11 a) ALB
ROSU
VERDE
b) 1, 2 si 3
c) y
d) "ROSU","ALB","VERDE"
e) executia programului este oprita prin STOP (linia 100)

CAPITOLUL 5
INTRARI, IESIRI, APLICATII SIMPLE
LET;INPUT;READ;PRINT;PRINT TAB;PRINT AT;REM;SAVE;LOAD

In acest capitol cunostintele dvs. de BASIC vor fi imbunatatite, prin detalii asupra posibilitatilor de introducere a datelor, de obtinere a rezultatelor si prin exemple de programe comentate.

Treptat, va veti insusi notiuni despre realizarea unor programe proprii bine gindite si clare, despre stocarea lor pe caseta magnetica si despre o operatie pe care o veti executa foarte frecvent si anume incarcarea unor jocuri/programe de pe casete magnetice.

1. Anulati programul existent in memorie. Aveti doua posibilitati:
   - -prin actionarea tastei RESET
   - -prin instructiunea NEW
2. Introduceti si executati urmatorul program:

```
10INPUT "a=";a
20INPUT "b=";b
30INPUT "c=";c
40LET d=a+b+c
50PRINT d
60STOP
RUN
```

Programul dispare de pe ecran si cursorul clipeste intrebator in coltul ecranului asteptind o valoare pentru variabila numerica "a".

Introduceti, de exemplu:

```
a=2
b=4
c=6
```

Scrieti ce se afiseaza:

____

Retineti ca cele trei instructiuni INPUT au permis introducerea prin interogare, de valori pentru variabilele numerice a, b si c.

3. Anulati liniile 10 si 20 si introduceti din nou comanda de executie

```
10  <ENT>
20  <ENT>
RUN <ENT>
c=9
```

Notati mesajul afisat: ____________________

CIP-ul a incercat fara succes sa execute adunarea din linia 40 si a semnalat ca nu a gasit variabilele "a" si "b".

4.Schimbati linia 30 punind toate variabilele, separate prin virgule:

```
30 INPUT "a=";a,"b=";b,"c=";c
```

i. Aveti programul:

```
LIST
30 INPUT "a=";a,"b=";b,"c=";c
40 LET d=a+b+c
50 PRINT d
60 STOP
RUN
```

De aceasta data cursorul clipeste cerind valoarea lui "a", dupa care sare la mijlocul liniei si cere pe "b" si apoi sare in linia urmatoare cerind pe "c".

```
a=10                b=4.3
c=5.7
Rezultat:_____________________
```

5.Introduceti:

```
30 READ a,b,c (READ obtineti cu SS+CS si apoi A)
LIST
```

Veti avea pe ecran:

```
30 READ a, b, c
40 LET d=a+b+c
50 PRINT d
60 STOP
RUN
```

Notati mesajul afisat: ________________________

Explicatia este urmatoarea: instructiunea READ este o alta posibilitate de a introduce date, direct in program sub forma de constante numerice fixe.

Deocamdata neavind aceste date, executia liniei 30 READ... a generat mesajul de mai sus, care va atrage atentia ca nu sint date.

6.Acum introduceti:

```
35 DATA 2, 4, 6
RUN
```

Avind datele cunoscute, CIP-ul executa fara ezitare adunarea si va comunica rezultatul.

7.Anulati linia 35 si introduceti linia 10:

```
35  <ENT>
10 DATA 2,4,6
<ENT>
10 DATA 2,4,6
30 READ a,b,c
40 LET d=a+b+c
50 PRINT d
60 STOP
RUN
```

Observati ca desi linia DATA a fost plasata in alta parte in program (chiar si inaintea instructiunii READ) BASIC-S accepta datele si executa programul.

8.Anulati programul anterior si introduceti:

```
10 READ x,y
20 LET z=x/y
30 PRINT z
40 GO TO 10
50 DATA 10,2,9,3,40,4
60 STOP
RUN
```

Notati ce se afiseaza: ___
___
___
____

Instructiunea READ a "citit" cite doua valori din DATA, in linia 20 a facut impartirea, linia 30 a determinat afisarea rezultatului si totul a fost reluat cu alta pereche de date pina cind acestea au fost epuizate.

9.Anulati linia 50 si introduceti:

```
5 DATA 36,6
45 DATA 64,32
70 DATA 35,5
RUN
Se afiseaza:
6
2
7
Out of DATA,10:1
```

Puteti pune deci mai multe instructiuni DATA intr-un program? (DA/NU)

Probabil ca ati raspuns corect: da! pot fi mai multe instructiuni DATA, amplasate oriunde in program.

10.Anulati programul si introduceti:

```
100 LET G=15
110 PRINT G
120 STOP
RUN
```

Notati rezultatul:

11.Schimbati linia 110:

```
110 PRINT "G"
RUN
```

Ce se afiseaza acum ca rezultat? __________

De retinut:

-PRINT constanta numerica - afiseaza valoarea constantei
-PRINT "constanta sir" - afiseaza continutul constantei sir.

12.Daca introduceti:

```
110 PRINT "greutatea=";G
```

Ce credeti ca se va afisa dupa executie? ______________________

Executati programul si verificati raspunsul dvs.

13.Anulati programul anterior si introduceti:

```
10 READ x
20 PRINT x
30 GO TO 10
40 DATA 20,21,22,23
```

Executati programul si notati rezultatul:

14.Schimbati:

```
20 PRINT x,
```

Observati ca apare o virgula dupa x. Executati programul si notati din nou rezultatul: _________ __________

_________ __________

15. Mai schimbati linia 20 astfel:

```
20 PRINT x;   (; dupa x)
RUN
```

Notati modul de afisare: _______

Retineti deci modul de afisare diferit dupa cum este incheiata instructiunea PRINT:

- -cu virgula: se afiseaza pe doua coloane (la marginea si la mijlocul ecranului);
- -cu punct si virgula: se afiseaza rezultatele unul alaturi de celalalt;
- -fara virgula/punct si virgula - se afiseaza rezultatele unul sub altul.

16. Adaugati si schimbati in program:

```
5 PRINT "01234567890123456789012345678901"
20 PRINT TAB 3;x
RUN
```

.Linia 5 a numerotat cele 32 (0-31) pozitii ale unei linii de afisare;

.In linia 20 s-a introdus TAB 3, care impune afisarea rezultatelor incepind cu pozitia 3.

Observati pe ecran obtinerea celor de mai sus.

17. Sa schimbam pozitia de afisare si datele:

```
20 PRINT TAB 15;x
40 DATA 8,100,3210,9.6
```

Rezultatele sint afisate incepind cu coloana 15.

18. Adaugati si schimbati in program:

```
8 PRINT 1:PRINT 2 - numeroteaza la margine linia 1 si linia 2
10 READ x$
20 PRINT AT 2,10;FLASH1;x$
      |            |
      |            └──► afiseaza clipitor
      └──► afiseaza incepind cu linia  2, pozitia (coloana) 10.
30 DATA "abc"
40 <ENT> - pentru a sterge linia 40.
```

Programul trebuie sa arate astfel:

```
5 PRINT "01234567890123456789012345678901"
8 PRINT 1:PRINT 2
10 READ x$
20 PRINT AT 2,10:FLASH1;x$
30 DATA "abc"
```

Executind programul veti obtine urmatorul rezultat:

```
          ▼── pozitia (coloana) 10
01234567890123456789012345678901
1
2 ◄───┐     abc ◄───┐
      |             |
   linia 2      clipitor
```

Notiuni de baza

Datele care sint primite de un program si prelucrate intr-un anumit fel, sint denumite DATE DE INTRARE.

BASIC-S permite trei moduri diferite de introducere a datelor, folosind instructiunile: LET, INPUT, READ..., DATA...

LET

Instructiunea LET atribuie unei variabile "v", valoarea unei expresii "e".

Sintaxa comenzii: LET v=e

Exemplu: LET a=15 se citeste "fie a egal cu 15" si de fapt atribuie variabilei "a" valoarea 15.

Exemple de instructiuni LET:

```
LET d=14.3
LET L$ = "APRILIE"
LET x=(p+m)/n
LET k=k+1
```

Ultimul exemplu pare imposibil dar il puteti intelege analizind urmatorul program:

```
┌──▶10 LET k=5
│   20 PRINT k;
│   30 LET k=k+1
└───40 GO TO 20
```

Dupa executie se afiseaza:

5678910...

Initial k ia valoarea 5 care este afisata. La executia liniei 30, valoarea anterioara este crescuta cu 1, astfel incit a doua oara va fi afisata valoarea 6 etc.

INPUT

Instructiunea INPUT (in traducere: introduceti) permite introducerea datelor direct de la tastatura.

Sintaxa: INPUT "c";v1;v2

.c este un sir optional de caractere care expliciteaza datele;

.v1,v2... sint nume de variabile numerice/sir de caractere.

Spre deosebire de LET, instructiunea INPUT permite deci introducerea conversationala a datelor. Cind CIP-ul o intilneste, opreste executia programului si cursorul clipeste intrebator pina cind dvs. raspundeti prin:

a) introducerea datelor solicitate, dupa care executia programului continua;

b) introducerea instructiunii STOP in vederea renuntarii la executie.

Exemple:

INPUT x - introduce pentru variabila x, valoarea tastata.

INPUT "a=?";a - afiseaza a=? si introduce valoarea tastata;

INPUT a;b;c - solicita pe rind valori pentru a, b si c.

Puteti deci cere introducerea datelor pentru mai multe variabile printr-o singura instructiune INPUT.

Variabilele pot fi separate prin punct si virgula sau prin virgula:

INPUT m;n - cere date pentru m si cursorul ramine in aceiasi pozitie.

INPUT m,n - cere date pentru m si apoi sare in coloana 15 pentru n.

READ...

Instructiunea READ (in traducere: citeste!) atribuie unor variabile v1, v2,... valoarea unor expresii/date succesive din listele instructiunilor DATA
Sintaxa: READ v1, v2,...

DATA

Instructiunea DATA permite introducerea valorilor pentru variabile.
Sintaxa: DATA c1, c2,...
Exemple:

```
READ m,n,p$,q$
DATA 25,1703,"cuvint","CIP"
```

Pentru a intelege mai bine ce efect au cele doua instructiuni, incercati exemplul de mai sus astfel:

```
10 DATA 25,1703,"cuvint","CIP"
20 READ m,n,p$,q$
30 PRINT m,n,p$,q$
RUN
```

Instructiunea READ difera de INPUT prin faptul ca programul nu se opreste sa astepte ca datele sa fie introduse prin tastatura, ci le cauta automat in instructiunea DATA. Instructiunea DATA poate fi plasata oriunde in program, chiar si inaintea instructiunii READ.

Se obisnuieste ca toate instructiunile DATA sa fie puse la sfirsitul programului, inaintea instructiunii STOP.

Pot fi mai multe instructiuni DATA si nu este necesar sa fie grupate in program. Instructiunea READ citeste pe rind valori pentru variabilele specificate, din DATA incepind cu numarul de linie cel mai mic. Daca numarul de variabile depaseste pe cel al datelor se afiseaza mesajul: Out of DATA si programul este oprit.

RESTORE

Instructiunea RESTORE (in traducere: restabileste!) permite folosirea aceluiasi set de date, de catre mai multe instructiuni READ.

Sintaxa: RESTORE␣n

Un indicator de citire este restabilit la instructiunea DATA cu numar de linie n, astfel incit urmatoarea instructiune READ va incepe citirea de la acea linie.
Exemplu:

```
10 READ a,b,c - citeste valori din liniile 50  si 60
20 RESTORE 50 - restabileste ca linie de citire linia 50
30 READ z,x   - va  citi  din  nou  in  linia 50
40 PRINT a'b'c'z'x' (introduceti apostrof intre variabile)
50 DATA 1,2
60 DATA 3,4
70 DATA 5,6 -  de aici nu citeste
80 STOP
RUN
```

Rezultat:

```
1
2
3
1
2
```

Nota: Separarea variabilelor din instructiunea PRINT cu apostrof, are ca efect afisarea fiecareia intr-o linie noua.

PRINT

Afiseaza pe ecran constante, variabile sau expresii calculate.
Sintaxa:

```
PRINT n       - afiseaza valoarea constantei numerice
PRINT "sir"   - afiseaza sirul dintre ghilimele
PRINT x;y;a$  - afiseaza x, y si a$ in aceeasi linie (fara spatii
                intre ele)
PRINT x,y,a$  - afiseaza alternativ incepind din coloanele 0 si 16
PRINT x'y'a$  - afiseaza fiecare valoare in linie noua.
```

Se pot face calcule cu ajutorul instructiunii PRINT. De exemplu:

```
PRINT x*y,z
```

va afisa din coloana 0 rezultatul inmultirii dintre x si y si din coloana 16 valoarea lui z.

Incercati:

```
NEW
10 LET x=5:LET y=6:LET z=7
20 PRINT "5*6=";x*y,z
RUN
```

Notati ce se afiseaza: ____________ ______

Schimbati si adaugati liniile:

```
20 PRINT x,y
25 PRINT ◄────── afiseaza o linie vida
30 PRINT x;y
35 PRINT
40 PRINTx'y
RUN

5          6 ◄──separare cu virgula
56           ◄──separare cu punct si virgula
5 }          ◄──separare cu apostrof
6 }
```

PRINT AT 1,c

Forma PRINT AT ... (in traducere afiseaza la ...) o puteti folosi cind vreti sa afisati ceva intr-o anume zona a ecranului. Afisarea incepe cu pozitia data de:

1 (0-21) - numar linie
c (0-31) - numar coloana

Incercati sa afisati in mijlocul ecranului un mesaj:

```
PRINT AT 11,9;"OPRITI BANDA"
          ▲ ▲
linia 11──┘ └──incepind din coloana 9
```

Afisarea pe ecran poate fi programata in doua moduri:
-grafica - pentru desene (va fi explicata in capitolul 9)
-alfanumerica - pentru caractere

Macheta ecranului

In afisarea alfanumerica ecranul este impartit in 24 linii orizontale si 32 coloane (Fig.5.1).

Afisarea unui caracter poate fi pozitionata oriunde pe ecran, prin doua coordonate: numar linie, numar coloana.

Liniile 22 si 23 sint folosite pentru editarea liniilor noi de program sau a celor in care doriti sa faceti modificari:

Incercati urmatoarea linie multipla PRINT care va produce un dreptunghi in mijlocul ecranului, in care va fi scris sirul "CIP":

```
    tastati:CS+9 si CS+8 ↓               ↓retastati CS+9 si apoi SS+P
    10 PRINT AT 7,11;"■■■■■■■■";AT 8,11;"■␣␣␣␣␣␣■";
       AT 9,11:"■␣CIP␣■";AT 10,11;"■␣␣␣␣␣␣■";AT11,11;
       "■■■■■■■■"
```

Caracterul ■ il obtineti cu (CS+9) CS+8 (Anexa B).

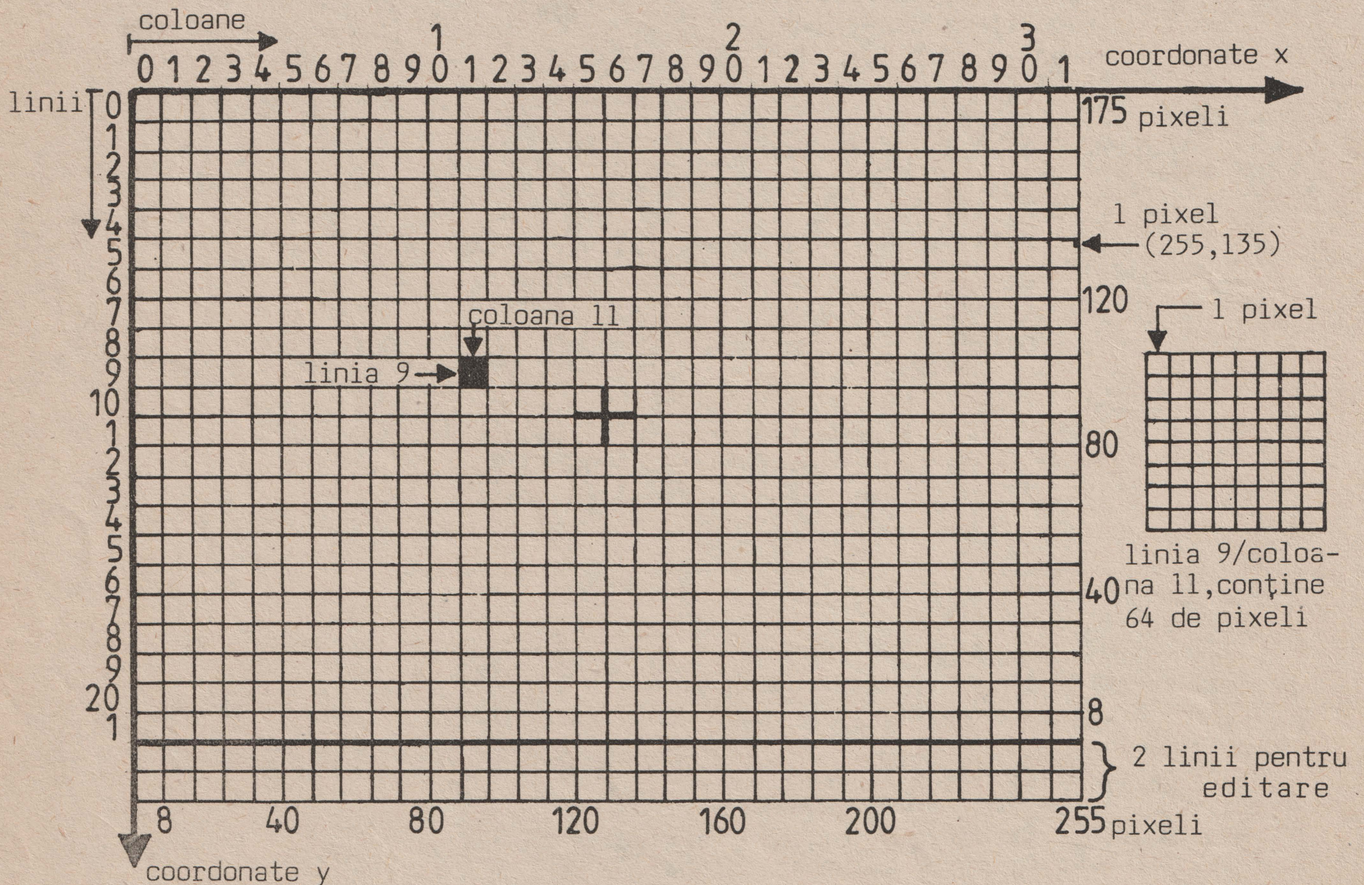


Fig.5.1

PRINT TAB

Pozitionarea pe liniile ecranului se mai poate obtine si cu forma

PRINT TAB c

unde c reprezinta coloana din care sa inceapa afisarea. Daca c este mai mare decit 31 afisarea se face in continuare pe linia urmatoare.

Introduceti si executati:

```
NEW
10  PRINT "01234567890123456789012345678901"
20  PRINT TAB 5;"♠";TAB 8;"♠";TAB 25;"♠";TAB 32;"*";TAB 35; "*"
RUN
```

REM

Instructiunea REM (prescurtare de la REMARK) permite introducerea in program a unor comentarii utile celui care analizeaza un program. REM nu este luata in considerare la executia programului, dar apare ori de cite ori programul este listat, usurind foarte mult intelegerea lui.

Este bine sa va obisnuiti sa introduceti asemenea comentarii in programele dvs.

Pentru a arata utilizarea instructiunii REM si avantajele ei, comparati cele doua exemple care urmeaza. Ambele programe produc aceleasi rezultate, dar unul din ele este "comentat".

```
 5 LET w=0
10 INPUT x,y,z
20 LET w=(x+y+z)/3
30 PRINT w
40 STOP
```

```
10 REM nume program:PRG
20 REM calculeaza media aritmetica
30 REM introducerea numerelor
40 INPUT x,y,z
50 REM calculul mediei
60 LET w=(x+y+z)/3
70 REM afisarea rezultatului
80 PRINT w
90 STOP
```

Exemple de programe

Stimate cititor, consider ca sinteti suficient de pregatit sa incepeti sa analizati si sa scrieti programe din ce in ce mai complicate.

In acest capitol vom analiza citeva exemple, vom enunta altele pe care sa le realizati singuri.

Curaj! Veti constata ca ati retinut mai multe cunostinte decit va inchipuiti.

Mai intii sa recapitulam:

- -un program este format din instructiuni
- -o linie de program poate contine una sau mai multe instructiuni separate prin":"
- -fiecare linie trebuie numerotata
- -instructiunea REM se foloseste pentru comentarea (documentarea) programului
- -pentru introducerea datelor se folosesc instructiunile:
  - .LET
  - .INPUT

.READ...DATA (RESTORE)
-pentru calcul se folosesc instructiunile:
.LET
.PRINT
-pentru afisarea rezultatelor se folosesc instructiunile:
.PRINT
.PRINT AT
.PRINT TAB

Iar acum citeva sugestii, pentru ca sa realizati programe bune si...frumoase. Este indicat sa urmati urmatoarele faze:

-analizarea problemei in vederea stabilirii:
.intrarilor
.prelucrarilor
.iesirilor
-proiectarea programului
-codificarea in limbaj BASIC
-testarea si punerea la punct a programului.

Stiu ca sinteti nerabdator sa faceti cit mai repede programe, dar nu sariti rindurile urmatoare prin care va semnalez citeva greseli facute de incepatori. Unele greseli sint simple, cele facute chiar la introducere (tastare gresita, erori de sintaxa) si vrind nevrind va trebui sa le corectati pentru ca CIP-ul vigilent nu le accepta.

Sint insa erori mai subtile, de logica, si acestea sint sesizabile de abia prin obtinerea unor rezultate necorespunzatoare. Pentru ele trebuie sa faceti o "depanare" a programului, actiune despre care vom discuta in alt capitol.

Pentru evitarea greselilor de introducere va recomand:

-respectati sintaxa ceruta de BASIC (in anexele C si D aveti sub forma de MEMENTO, toate comenzile / instructiunile in ordine alfabetica).
-tastati <ENT> la sfirsitul fiecarei linii de program.
-nu tastati litera O in loc de cifra 0 (zero).
-fiti atent la numarul de linie, sa nu il repetati pe cel al uneia existente, pentru ca aceasta va dispare.
-anulati vechiul program prin NEW sau RESET!

Exemplul 5.1

Sa se realizeze un program pentru calculul unei expresii matematice date:
Intrari:
a,b,c,m,n - variabile numerice introduse prin tastatura in coloanele 0 si 16.

Prelucrari: $x = \left[2(a+b)^2 + 3c^2 \right]^{\frac{m}{n+1}}$

Iesiri:
x= valoarea lui x
Se doreste deci nu numai afisarea valorii rezultatului, ci a unei forme explicite
x=rezultat
Algoritmul problemei:
1.solicitarea si memorarea datelor de intrare.
2.calculul expresiei
3.afisarea rezultatului
4.stop

Codificarea programului:

```
10 REM nume program PRG5.1
20 REM program pentru calculul unei expresii matematice
```

```
30 REM introducerea datelor
40 INPUT a;b;c;m;n
50 REM calculul expresiei
60 LET x=(2*(a+b)↑2+3*c↑2)↑(m/n+1)
70 REM afisarea rezultatului
80 PRINT "x=";x
90 STOP
```

Doua precizari:

1.-la expresii cu paranteze trebuie sa verificati ca numarul parantezelor deschise "(" sa fie egal cu cel al parantezelor inchise ")".

2.-daca va aflati in executia unei instructiuni INPUT si dintr-un motiv oarecare doriti sa intrerupeti introducerea, singura posibilitate este de a sterge inapoi cu DELETE pina se opreste cursorul si sa dati STOP <ENT>.

Introduceti programul, notati pe o hirtie datele introduse si rezultatul: x=_________

Schimbati linia 60 astfel:

```
60 LET x=(2*(a+b)↑2+3*c↑2)↑(m/(n+1))
```

Executati programul cu aceleasi date si notati din nou rezultatul x=_______.

Observati ca rezultatele difera desi nu v-a fost semnalata nici o eroare la introducere.

De ce difera rezultatele?

Care este cel bun? (nu calculati cu creionul, ci analizati programul!)

Exemplul 5.2

Sa se scrie un program pentru calculul sumei totale a unor cantitati date.

- Intrari:
  c=cantitatea, ceruta sub forma explicita:
  cantitate = valoarea ce se introduce
- Iesiri:
  total - suma cantitatilor introduse pina in acel moment.
  Afisarea sa se faca in doua rubrici denumite CANTITATEA si TOTAL decalate astfel:

```
          ▼   coloana 0           ▼    coloana 16
          CANTITATEA              TOTAL
          ==========              ===== ◄— subliniere
de ex.        3                         ◄— prima linie cu date
              2                     3   ◄— pe alta linie
                                    5   ◄— (3+2)
```

-Prelucrari

.TOTAL = 0 - valoarea initiala a variabilei TOTAL

.TOTAL = TOTAL + c - se insumeaza totalul anterior cu cantitatea nou introdusa

-Codificarea programului:

```
10 REM nume program PRG5.2
20 REM program pentru insumarea unor date
30 REM afisarea capului de tabel
40 PRINT "CANTITATEA","TOTAL"
50 PRINT "==========","====="
60 REM solicitarea cantitatii
70 INPUT "cantitate=";c
```

```
80 REM calculul totalului
90 LET total = total + c
100 REM afisarea cantitatii si a totalului
110 PRINT c',total
120 REM solicitarea altei cantitati
130 GO TO 70
RUN
```

Ce s-a intimplat?

A fost afisat capul de tabel, dar sinteti avertizati ca nu a fost gasita variabila in linia 90. Deoarece variabila c tocmai ati introdus-o rezulta ca nu a fost gasita variabila total.

CIP-ul "v-a prins" cu o greseala de logica. I-ati cerut ceva pe care ati uitat sa i-o definiti mai inainte.

Corectati greseala introducind ceea ce de fapt ati prevazut la prelucrari: declararea si initializarea variabilei "TOTAL".

```
25 REM initializarea variabilei de totalizare
28 LET total = 0
RUN
```

Acum programul se executa. Introduceti citeva cantitati si verificati:

```
CANTITATE             TOTAL
=========             =====
   4
                        4
   6
                       10
 185
                      195
1000.5
                     1195.5
  2E+9
              2.0000012E+9
```

La ultimul total, numarul fiind foarte mare, a fost rotunjit pentru a putea fi afisat cu 8 cifre:

2.0000011955 ⟶ 2.0000012

Salvarea programelor pe caseta

Deoarece in continuare va voi propune sa realizati singuri programe pe "teme" date, veti dori eventual sa le pastrati si dupa oprirea calculatorului ca sa le revedeti altadata.

Pentru aceasta va fi nevoie sa le salvati pe o caseta magnetica (memoria externa a CIP-ului), de unde sa le incarcati din nou in memorie oricind doriti.

Pregatiti deci casetofonul, o caseta (atentie, ce veti inregistra va sterge ceea ce aveati deja pe caseta!).

Puneti cablul de legatura intre CIP si casetofon.

Inregistrarea programelor pe caseta

Cind doriti sa salvati un program procedati astfel:

a) - fiind in mod comanda (cursor K) introduceti comanda:

```
SAVE"nume program" LINE numar <ENT>
```

Exemplu:

SAVE "PRG5.2" LINE 10 <ENT>
Obtineti mesajul: Start tape, than press any key
adica: Porniti banda, apoi apasati orice tasta

b) - fara sa porniti banda apasati orice tasta.

L-ati "pacalit" pe CIP, el incepe transmiterea programului catre casetofon si puteti astfel regla nivelul de inregistrare. Acesta este bine sa fie cit mai mare, chiar peste limita recomandata pentru muzica.

c) - mai introduceti odata comanda
SAVE "nume program" LINE numar <ENT>
si acum ascultati-i sfatul, pornind intii banda si dupa citeva secunde (pentru a se crea un spatiu pe banda intre programe) apasati orice tasta. Daca totul merge normal obtineti confirmarea prin OK (totul in ordine).

Un program salvat numai cu:

SAVE "nume program" <ENT>

la incarcarea de pe caseta va fi lansat in executie prin comanda RUN, spre deosebire de cazul in care introduceti in comanda SAVE numarul primei linii din program (ca in exemplul de mai sus), cind programul va fi lansat automat in executie.

Alte forme ale comenzii SAVE le gasiti in anexa C.

VERIFY

O deprindere buna este aceea de a verifica de fiecare data, daca salvarea s-a facut corect. Pentru aceasta:

a) - derulati banda la inceputul inregistrarii

b) - introduceti comanda:

VERIFY "nume program" <ENT>

c) - porniti banda

Daca programul a fost inregistrat corect, pe ecran apare:
Program nume
si la sfirsitul verificarii se afiseaza
0 OK

Mesajul

R Tape loading error
indica o eroare de inregistrare

Incarcarea programelor de pe caseta

Asa cum v-am mai spus CIP-ul prezinta un mare avantaj, acela de a putea executa orice program elaborat pentru HC 85/TIM-S/SINCLAIR-SPECTRUM.

Programele se difuzeaza pe casete, asa incit veti fi frecvent in situatia de a avea niste jocuri sau programe formidabile pe o caseta si veti fi nerabdatori sa le executati pe CIP-ul dvs.

Pentru incarcarea unui program de pe caseta procedati astfel:

a) - introduceti

LOAD "" - daca doriti sa fie incarcat primul program care urmeaza pe banda

LOAD "nume program" - pentru a fi cautat si incarcat programul cu numele specificat in comanda

b) - porniti banda

Daca totul este in ordine, pe ecran se afiseaza numele programului si urmeaza dungile clipitoare insotite de un sunet specific.

c) - opriti banda la aparitia mesajului "OK" sau a celui dat de program.

Daca dupa pornirea benzii nu apar numele programului si dungile orizontale, sau daca apare mesajul: "Tape loading error" opriti banda, derulati-o rapid inapoi si reluati secventa de incarcare. Dupa 2-3 incercari nereusite, puteti renunta la incarcarea acelui program, deoarece poate fi una

din urmatoarele cauze:

-programul a fost salvat cu un nivel de inregistrare necorespunzator;

-casetofonul dvs. nu are capul de citire aliniat la fel cu cel pe care s-a facut salvarea;

-banda este deformata (intinsa, cutata) sau zgiriata.

O ultima solutie pe care o puteti incerca este aceea de a utiliza alt casetofon.

Intrebari recapitulative si exercitii

I 5.1 Ce se va afisa daca se executa urmatorul program?

```
10 LET x=1
20 PRINT x,
30 LET x=x+1
40 GO TO 10
```

Scrieti primele 12 valori rezultate si apoi executati programul pentru a va verifica (intrerupeti cu BREAK).

I 5.2 Care este scopul instructiunii REM?

I 5.3 Daca introduceti intr-un program instructiunea READ, ce alta instructiune trebuie sa utilizati obligatoriu?

I 5.4 Ce va fi afisat daca se executa urmatorul program?

```
10 LET a=6:LET b=13:PRINT"b=";a
```

I 5.5 Completati valorile lipsa:

In afisarea alfanumerica ecranul este considerat impartit in _____ coloane si _____ linii.

I 5.6 Ce instructiune folositi pentru a afisa incepind din diferite coloane?

I 5.7 Difera afisarea rezultatelor prin liniile 30,40 si 50 din urmatorul program?

```
10 LET a=7
20 LET b=4
30 PRINT a
40 PRINT b
50 PRINT a'b
```

I 5.8 Ce tastati pentru a afisa programul daca CIP-ul asteapta un raspuns dupa o instructiune INPUT?

I 5.9 Ce instructiune permite utilizarea aceluiasi set de date, de catre mai multe instructiuni READ?

I 5.10 Scrieti in format BASIC expresiile:

a) $2(x_1 + 3x_2)$

b) $x = \dfrac{5,4 + y\sqrt{u+v}}{(a+b)^{2/y}}$

I 5.11 Scrieti instructiunile BASIC pentru:

a) a creste c7 cu 0,01

b) a da valoarea "**eroare**" variabilelor T$ si U$.

I 5.12 Ce erori exista in urmatorul program:

```
10 INPUT l,h
20 a=l*h
30 PRINT l,h,a,p
40 LET p=2(l+h)
```

Probleme

1) Scrieti un program pentru calculul suprafetei (S) si al volumului (V) unei sfere cu raza r. Rezultatele sa fie afisate astfel:

r=____␣S=______␣V=______␣

2) Scrieti un program pentru calculul volumului unei cutii paralelipipedice avind lungimea L, latimea l si inaltimea h.

Datele de intrare sa fie cerute astfel:

Lungimea (cm)?
Latimea (cm)?
Inaltimea (cm)?

Rezultatele sa fie afisate astfel:

L =   cm
l =   cm
h =   cm
Volumul este   centimetri cubi.

Raspunsuri

R 5.1

1	2
3	4
5	6
7	8
9	10
11	12

R 5.2 Instructiunea REM permite introducerea unor comentarii care sa ajute la intelegerea programului.

R 5.3 DATA

R 5.4 b=6

R 5.5 32 coloane (0-31)
22 linii (0-21)

R 5.6 PRINT TAB

R 5.7 Nu! Apostroful din linia 50 provoaca afisarea rezultatelor la inceput de rind, unul sub altul.

R 5.8
1. STOP <ENT>
2. <ENT>

R 5.9 RESTORE

R 5.10
a) 2*(x1 + 3*x2)
b) x=(5.4 + y*(u+v)↑0.5)/(a+b)↑2/y

R 5.11
a) LET C7 = C7 + 0.01
b) LET T$ = "**eroare**"
LET U$ = "**eroare**"

R 5.12
- in linia 20 lipseste LET
- linia 30 trebuie sa devina 50 pentru ca p nu este cunoscuta
- corect se scrie: LET p = 2*(l+h)

CAPITOLUL 6
CUM INTRODUCEM CONDITII SAU ALTERNATIVE IN PROGRAM?

Adeseori veti simti nevoia in programe sa determinati CIP-ul sa nu mai execute instructiunea care urmeaza, ci sa execute un salt la o instructiune aflata peste un numar de linii inainte sau inapoi.

Un asemenea salt poate fi introdus in program, sa fie executat in anumite conditii sau neconditionat.

```
IF...THEN
GO TO
```

Un utilizator necunoscator in programarea calculatoarelor va avea impresia ca CIP-ul ia "decizii" in anumite momente, dar de fapt acestea sint introduse de autorul programului/jocului sub forma unor alternative conditionate sau nu.

1.Introduceti urmatorul program:

```
NEW
110 LET a=1
120 PRINT a
130 LET a=a+1
140 IF a<8 THEN GO TO 120
150 STOP
```

Linia 140 o cititi astfel:

"DACA a este mai mic decit 8 ATUNCI MERGI (sari) la linia 120"

Analizati cu atentie programul si scrieti pe o hirtie ce credeti ca va fi afisat.

Executati programul si verificati-va.

2.Schimbati linia 110:

```
110 LET a=6
```

Ce se va obtine?

Executati programul si verificati!

3. Schimbati si introduceti:

```
130 LET a = a + 10
140 IF a>=46 THEN GO TO 150
145 GO TO 120
RUN
```

Acum linia 140 a comandat CIP-ului:

"DACA a este mai mare decit sau egal cu 46 ATUNCI MERGI la linia 150".

CIP-ul ascultator asa va face, dar cit timp a este mai mic decit 46 va executa linia urmatoare (145) care il trimite inapoi la linia 120.

Analizati programul si rezultatele pentru a intelege ce s-a intimplat.

4. Completati semnele prin care veti pune in program conditia pentru (v.capitolul 4):

mai mare decit _____
mai mic decit sau egal cu _____
egal cu _______
mai mare decit sau egal cu _____
neegal cu (diferit de) ________

5. Analizati cu atentie urmatorul program:

```
10 PRINT "introduceti 2, 4 sau 6"
20 INPUT "n=";n
30 IF n=2 THEN GO TO 90
40 IF n=4 THEN GO TO 70
50 IF n=6 THEN GO TO 80
60 GO TO 100
70 PRINT "SCUFITA ROSIE": GO TO 20
80 PRINT "HARAP ALB": GO TO 20
90 PRINT "PINOCHIO": GO TO 20
100 STOP
```

Ce valoare trebuie sa introduceti pentru ca sa se afiseze:

```
HARAP ALB        n=______
PINOCHIO         n=_______
SCUFITA ROSIE    n=_______
```

Anulati vechiul program, introduceti pe cel de mai sus si executati-l cu valorile anticipate de dvs. verificindu-va!

6. Programul cere sa se introduca pentru n valorile 2, 4, 6. Ce credeti ca se va intimpla daca introduceti alta valoare? Formulati raspunsul si verificati!

Notiuni de baza

GO TO

Instructiunea GOTO produce un salt neconditionat la linia indicata.

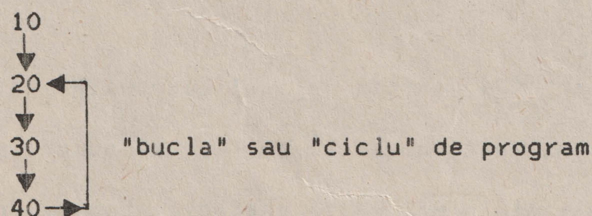
Sintaxa: GO TO numar linie

Fie de exemplu urmatorul program:

```
10 LET x=5
20 PRINT x
30 LET x=2*x
40 GO TO 20
50 STOP
```

Un program BASIC este executat incepind cu linia avind cel mai mic numar si continua linie cu linie. Instructiunea GO TO (in traducere: "mergi la") permite intreruperea acestei secvente printr-un transfer /salt la o linie de program specificata.

Programul de mai sus se executa in urmatoarea ordine a liniilor:



Deoarece nu este pusa nici o conditie in linia 40, GO TO constituie o instructiune de salt neconditionat.

Linia 40 produce saltul executiei inapoi la linia 20, astfel ca linia 50 de oprire a programului nu ar fi executata niciodata.

Intrucit programul nu se opreste singur, se spune ca "a intrat intr-o bucla fara sfirsit" sau "cicleaza".

Intreruperea executiei se poate face dupa cum deja stiti prin:
-STOP - la intrebarea data de INPUT
-BREAK

IF...THEN

Instructiunea IF...THEN (in traducere: "daca...atunci") permite introducerea in program a unui transfer (salt) conditionat.

Sintaxa: IF x THEN s

Daca expresia x este adevarata (conditia este indeplinita) se executa s, in caz contrar se executa instructiunea care urmeaza dupa IF.

Exemple:

```
IF j=3 THEN LET i=i+1:GO TO...
IF a>0 AND a<10 THEN GO TO (3+a*100)
IF ch=z OR z=66 AND ch<>66 THEN...
IF NOT a OR NOT w THEN...
IF 4*(x+5) <(a-b) THEN...
```

Sa consideram in acest ultim exemplu
x=1   a=50   b=25
La executie calculatorul ajunge sa verifice daca:
4 * (1 + 5) < (50 - 25)
24 < 25

In acest caz conditia este indeplinita si se va executa ceea ce urmeaza dupa THEN. Daca dupa calculul expresiilor rezulta o conditie neindeplinita, s-ar executa instructiunea cu numarul de linie imediat superior dupa IF.

Instructiunea de transfer conditionat IF THEN este deosebit de utila, dind posibilitatea calculatorului sa sara in program oriunde doriti.

Dupa cum poate ati remarcat, programele luate de exemplu pina acum, au avut o mare deficienta: atunci cind includeau cicluri de repetare a unui numar de linii, nu exista alta cale de a fi oprite decit prin STOP/BREAK/RESET introdus de dvs.

Saltul conditionat (cu IF...THEN) permite o solutie mai eleganta de a stopa un program in anumite conditii.

Exemple de programe

Exemplul 6.1

Sa se realizeze un program care sa afiseze incepind din coloanele 0 si 16, numerele:

```
 3    6
 9   12
15   18
21   24
```

Dupa ce ultimul numar (24) a fost afisat executia sa se opreasca automat.

Analizind putin problema data, observam ca fiecare numar este mai mare cu 3 decit cel anterior.

Programul poate fi realizat in mai multe variante: Cea mai putin eleganta este:

```
10 PRINT "3  (15 spatii)  6"
20 PRINT "9  (15 spatii)  12"
30 PRINT "15 (14 spatii)  18"
40 PRINT "21 (14 spatii)  24"
50 STOP
```

Poate unii zimbiti cu superioritate, dar retineti ca la un incepator este important sa faca un program care sa dea rezultatele cerute, nu neaparat sa fie cel mai bun program!

Nu este o solutie prea buna pentru ca cere multa atentie la introducere si nu permite usor o eventuala schimbare a numerelor, sau a spatiilor de afisare.

O alta solutia ar fi:

```
10 LET n=3
20 PRINT n,
30 LET n=n+3
40 IF n >= 25 THEN STOP
50 GO TO 20
RUN
```

Virgula din linia 20 a produs afisarea in coloanele 0 si 16, iar linia 40 a conditionat stoparea programului.

O a treia solutie se obtine schimbind:

```
40 IF n <= 24 THEN GO TO 20
50 STOP
```

Introduceti si executati fiecare varianta analizind apoi cu atentie programele si rezultatele.

Exemplul 6.2

Sa se realizeze un program care sa faca media aritmetica a unor numere.

-Analizarea problemei:

In tema data nu se specifica la cite numere trebuie facuta media. Daca este vorba de multe numere, cea mai buna forma de introducere va fi cu READ...DATA.

Pentru a putea stopa programul dupa ultimul numar (fara a sti care este), sa introducem un numar - indicator care sa semnaleze calculatorului ca este ultimul, si sa se opreasca programul.

Sa alegem ca indicator valoarea "0" pentru ca tot nu ar influenta media ce se calculeaza.

Trebuie deci pusa in program conditia ca atunci cind este citita valoarea 0 programul sa se opreasca.

Fie urmatoarele variabile:

x - un numar pentru calculul mediei
s - suma numerelor
n - numarul de numere pentru care s-a calculat s
m - media aritmetica

$$m = \frac{s}{n}$$

Codificarea programului:

```
 5 REM nume program: MEDAR
10 REM initializare s si n
20 LET s=0: LET n=0
30 REM citire date
40 READ x
50 REM testare sfirsit de date
60 IF x=0 THEN GO TO 110
65 PRINT x
70 LET s=s+x
80 LET n=n+1
```

```
 90 GO TO 40
100 REM calculul mediei
110 LET m=s/n
120 REM afisarea mediei
130 PRINT "------"
140 PRINT "media este:␣";m
150 DATA 6,3,7,5,4,0
160 STOP
    RUN


6
3
7
5
4
------
media este: 5
```

Un amanunt: la scrierea liniei 60 se lasa necompletat numarul liniei la care sa se faca saltul (GO TO...) pina cind se stie care este acea linie.

Se pot pune in program atitea instructiuni DATA cite sint necesare pentru a cuprinde toate numerele, avind grija sa introducem 0 dupa ultimul numar.

Intrebari recapitulative si exercitii

I 6.1 Ce se va afisa dupa executia urmatorului program:

```
10 LET a=3
20 LET b=2*a
30 PRINT a, b
40 LET a=a+2
50 IF a <= 10 THEN GO TO 20
60 STOP
```

I 6.2 Indicati greselile din urmatoarele instructiuni:

```
a) IF D↑3 >= 199 THEN 30
b) IF a2 = 5 THEN k=k+1
c) GO TO 90 IF t=7
d) IF q > w : GO TO 140
```

I 6.3 Scrieti instructiunile pentru urmatoarele conditii:

a) daca x < 3 afiseaza "negru", in caz contrar afiseaza "alb"
b) daca sirul A$ este "da" salt la 90, daca nu, salt la 200
c) daca t=0 salt la 110, daca nu, creste x cu 1 si salt la 40
d) daca x este cuprins intre 1 si 4, salt la linia (1000 + x)

Probleme

Scrieti programe BASIC:

1) Sa ceara doua numere si sa-l afiseze pe cel mai mare
2) Sa citeasca cinci numere in DATA si sa-l afiseze pe cel mai mic;
3) Sa calculeze si sa afiseze suma tuturor numerelor de la 1 la 50;
4) Sa ceara doua numere. Daca amindoua sint mai mari decit sau egale cu 10 sa se afiseze suma lor, iar daca amindoua sint mai mici ca 10 sa se afiseze produsul lor. Daca un numar este sub 10 si celalalt este mai mare sau egal cu 10, sa se afiseze diferenta dintre cel mai mare si cel mai mic.

Raspunsuri

R 6.1
```
3     6
5    10
7    14
9    18
```

R 6.2 a) lipsa GO TO dupa THEN
b) lipsa LET dupa THEN
c) IF si GO TO inversate si lipsa THEN
d) lipsa THEN

R 6.3 a)
```
50 IF x < 3 THEN PRINT "negru": GO TO 70
60 PRINT "alb"
70 ..........
```
b)
```
IF A$ = "da" THEN GO TO 90
GO TO 200
```
c)
```
IF t=0 THEN GO TO 110
LET x=x+i : GO TO 40
```
d)
```
IF x>=1 AND x<=4 THEN GO TO (1000 + x)
```

CAPITOLUL 7
CUM SE POT REPETA PARTI DIN PROGRAM

Veti invata in continuare o noua si foarte folositoare posibilitate a limbajului BASIC, de a realiza cicluri automate in cadrul programelor.

```
FOR
:
NEXT
```

In capitolul anterior v-am aratat cum se pot face repetitii/cicluri, folosind instructiuni de transfer conditionat sau neconditionat. BASIC are insa si instructiuni speciale care sa inceapa, sa execute si sa termine automat cicluri in program.

Aplicatie practica AP7

1. Introduceti:

```
NEW
10 LET a=2
20 PRINT a,: LET a=a+3
30 IF a <= 17 THEN GO TO 20
40 STOP
```

Notati rezultatele executiei:

```
-----      ------
-----      ------
-----      ------
```

2. Tastati:

```
NEW
10 FOR a=2 TO 17 STEP 3
20 PRINT a,
30 NEXT a
40 STOP
RUN
```

Remarcati ca se obtin aceleasi rezultate.

3. Schimbati linia 10:

```
10 FOR a=2 TO 20 STEP 2
RUN
```

Analizati rezultatele! Observati ca sint afisate din 2 in 2, adica cifra pusa la cuvintul STEP (in traducere: PAS)

4. Schimbati din nou:

```
10 FOR a=2 TO 10
   RUN
```

Nemaifiind specificat STEP, calculatorul da valori lui a cu un PAS egal cu 1.

5. Tastati:

```
10 FOR a=20 TO 10 STEP -2
RUN
```

Notati rezultatele:

------ ------
------ ------
------ ------

6. Incercati acum:

```
10 FOR a = 10 TO 20 STEP -2
RUN
```

Notati mesajul obtinut si incercati sa-l explicati:

----

7. Sa vedem ce se intimpla daca se folosesc doua variabile in cicluri FOR...NEXT.

Introduceti:

```
NEW
10 PRINT "a","b"
20 PRINT
30 FOR a=1 TO 3
40 FOR b=2 TO 5
50 PRINT a,b
60 NEXT b
65 PRINT
70 NEXT a
80 STOP
RUN
```

Notati rezultatele si sa incercam acum sa interpretam putin programul:

- -linia 10 afiseaza numele variabilelor
- -linia 20 introduce o linie vida
- -linia 30 se citeste: "pentru a egal cu 1 la 3" si deschide un ciclu pentru variabila a
- -linia 40 se citeste: "pentru b egal cu 2 la 5" si deschide un ciclu pentru variabila b
- -linia 60 se traduce: "urmatoarea valoare pentru b" si comanda reluarea ciclului variabilei b,cu linia 40.Pentru a=1,b ia succesiv valorile 2, 3, 4, 5 (priviti la rezultate)
- -linia 70 produce salt la linia 30 si trecerea variabilei a la valoarea urmatoare : a=2 reluindu-se ciclul variabilei b. Acelasi lucru se intimpla si pentru a=3 dupa care programul este stopat de linia 80.

8. Modificati:

```
40 FOR b=1 TO 2
```

Incercati sa scrieti ce se va obtine si apoi executati programul si confruntati rezultatele cu cele anticipate de dvs.

Notiuni de baza

```
FOR
.
.
.
NEXT
```

Instructiunile FOR...NEXT permit executia instructiunilor cuprinse intre ele de un numar limitat de ori.

```
Sintaxa:  FOR x=n TO m          sau
          FOR x=n TO m STEP s
          :
          NEXT x
```

- x este numele variabilei (o singura litera)
- m, n, s sint numere sau expresii aritmetice.

Exemple:

```
a)   ┌─►30 FOR x=2 TO 5 (se traduce: PENTRU x=2,3,4,5)
     │   .
     │   . linii de program care se vor putea repeta
     │   .
     └───170 NEXT x (se traduce: URMATOAREA valoare a lui x)
b)       FOR i=a TO a+b STEP d
         .
         .
         NEXT i
c)       FOR x = (abc - 2) TO 7 STEP -1
         FOR j=1 TO i
         .
         .
         NEXT j : NEXT x
d)       FOR i = 0 TO INT (6.9 * RND)
         .
         .
         NEXT i
```

Sa vedem putin mai in detaliu cum lucreaza instructiunile FOR...NEXT . Introduceti:

```
NEW
10 LET n=5
20 FOR i = 2 TO 4 ┐
30 LET n = n+1    ├   Se repeta pentru
40 PRINT i, n     │   i=2 i=3 i=4
50 NEXT i         ┘
60 PRINT : PRINT "****"
70 STOP
   RUN
```

Iata rezultatele dupa executia fiecarei linii:

```
nr. linie
 10  n=5
 20  i=2   ┌──►i=3   ┌──►i=4
 30  n=6   │   n=7   │   n=8
 40   ≀    │    ≀    │    ≀
 50   ▼    │    ▼    │    ▼
 60   └────┘    └────┘   ****
 70                      STOP
```

La inceput n primeste valoarea 5, apoi calculatorul intra in prima executie a ciclului, variabila i luind valoarea 2. Se executa liniile 30 si 40 cuprinse in ciclu si linia 50 care produce "saltul" executiei inapoi la linia 20 si i devine 3. Bucla se repeta pina cind i atinge valoarea limita 4 cu care calculatorul iese din ciclu si se executa instructiunea imediat urmatoare, adica linia 60.

Clauza STEP±s determina cresterea, respectiv scaderea variabilei cu "s" la fiecare executie a ciclului.

```
FOR x=0 TO 9 STEP 3
x va fi 0,3,6,9
FOR x=16 TO 0 STEP -4
x va fi 16,12,8,4,0
```

In program puteti introduce oricite cicluri FOR...NEXT, respectind urmatoarele conditii:

-un ciclu deschis cu FOR trebuie inchis neaparat cu NEXT,deoarece in caz contrar executia are loc o singura data, pentru prima valoare a lui x

-buclele sa nu se intersecteze

```
10 FOR a=──┐          10 FOR a=─┐
20 FOR b=┐ │          20 FOR b=─┼┐
-        │ │          -         ││
-        │ │corect    -         ││incorect
-        │ │          -         ││
60 NEXT b┘ │          60 NEXT a─┘│
70 NEXT a──┘          70 NEXT b──┘
```

Iata alt exemplu de bucle multiple organizate corect:

```
┌──────10 FOR x=10 TO 50
│  ┌─20 FOR y=6 to 8
│  │  -
│  │  -
│  │  -
│  └─100 NEXT y
│     -
│     -
│     -
│ ┌──150 FOR z=3 TO 7
│ │   -
│ │   -
│ │   -
│ │ ┌─200 FOR w=26 TO 12 STEP -2
│ │ │  -
│ │ │  -
│ │ │  -
│ │ └─250 NEXT w
│ │   -
│ │   -
│ └──320 NEXT z
│     -
│     -
└──────360 NEXT x
```

Exemple de programe

Exemplul 7.1

Program pentru afisarea numerelor cuprinse intre 10 si 50, din 5 in 5.

a)

```
NEW
10 LET a=10
20 PRINT a,
30 LET a=a+5
40 IF a<= 50 THEN GO TO 20
50 STOP
```

b)

```
10 FOR a=10 TO 50 STEP 5
```

```
20 PRINT a,
30 NEXT a
40 STOP
```

Introduceti programele (aveti grija! NEW inainte de al doilea) si executati-le verificind identitatea rezultatelor.

Exemplul 7.2

Sa reluam programul realizat la 6.2, pentru calculul mediei aritmetice a unor numere introduse de la tastatura.

-Analizarea problemei.

Se cere sa introducem numere la cerere (interactiv) si programul sa calculeze media lor.

Afisarea numerelor sa se faca la marginea ecranului, unul sub altul, si la sfirsit sub o linie de total sa se afiseze media.

-Variabile:
x - un numar pentru calculul mediei
s - suma numerelor
n - numarul de numere pentru care sa se calculeze s
m - media aritmetica : m = s / n

-Codificarea problemei:

```
5 REM nume program: MEDAR1
10 INPUT "cite numere?␣";n
20 LET s = 0
30 REM calculul sumei
40 FOR i = 1 TO n
50 INPUT "numarul?␣";x
60 LET s = s + x
70 PRINT x
80 NEXT i
90 REM calculul mediei aritmetice
100 LET m=s/n
110 PRINT "--------"
120 PRINT "media este:";m
130 STOP
```

Executati programul cu setul de numere din exemplul 6.2:
n=5 numere
x=6,3,7,5,4

Exemplul 7.3

Un program care sa va arate ca sinteti atit de bine pregatit, incit puteti descifra continutul memoriei calculatorului. Despre lucrul intim cu memoria principala veti afla in capitolul 12. Acum anticipam insa putin cu citeva notiuni.

Memoria principala o putem imagina ca un dulap cu multe sertare numerotate unul dupa altul, incepind cu sertarul avind numarul 0 si continuind din 1 in 1 pina la sfirsitul memoriei.

Intr-un asemenea "sertar" calculatorul memoreaza un caracter si numarul sertarului reprezinta o ADRESA.

Caracterele, va amintiti, sint reprezentate prin codurile cuprinse in anexa A.

Daca introduceti o linie de program:

```
PRINT "x+y=";5
```

ea este memorata astfel (verificati codurile din anexa A):

adresa	continut	caracter
23759	245	PRINT
23760	34	"
23761	120	x
23762	43	+
23763	121	y
23764	61	=
23765	34	"
23766	59	;
23767	53	5

Exista o instructiune care citeste continutul memoriei la orice adresa dorim si il afiseaza pe ecran:

PEEK a unde "a" reprezinta adresa

Iata un program care va afiseaza adresele de mai sus si continutul lor:

```
10 PRINT "x+y=";5
100 PRINT "adresa";TAB 7; "continut"
110 PRINT "======";TAB 7;"========"
120 REM ciclu pentru citirea si afisarea adreselor
130 FOR a = 23759 TO 23767
140 PRINT a; TAB 7; PEEK a
150 NEXT a
```

Executati programul incepind cu linia 100:

```
160 GO TO 100
```

Analizati rezultatele, asa-i ca este interesant!

Intrebari recapitulative si exercitii

I 7.1 Ce se va afisa dupa executia urmatoarelor programe:

a)
```
10 FOR a=20 TO 0 STEP -4
20 PRINT a,
30 NEXT a
40 STOP
```

b)
```
10 FOR a=2 TO 6 STEP 2
20 FOR b=2 TO 4
30 PRINT a;b;
40 PRINT a+b
50 NEXT b
60 NEXT a
70 STOP
```

I 7.2 Indicati greselile din urmatoarele linii:

a) FOR a ↑ 3 = n TO 120 STEP 5
b) FOR k$ = 1 TO 25
c) FOR x = y TO n STEP x
d) FOR a = 25 TO 70 STEP i
NEXT i

I 7.3 Scrieti buclele FOR...NEXT pentru:
a) o repetitie de 30 de ori, cu pas 2
b) x cuprins intre 1 si 10 si o repetitie cu pas 1 pina cind x > 6 si atunci sa se afiseze x si "abc"
c) o repetitie cu n cuprins intre 0,8 si (a^2+3) si cu pasul de (a+b)

Probleme

1. Scrieti un program pentru calculul si afisarea urmatoarelor numere ridicate la cub:

n	n^3
2,0	8
2,1	
2,2	
.	
.	
2,9	
3,0	27

2. Scrieti un program care sa citeasca in DATA numele a 5 elevi, notele obtinute de fiecare la 3 materii si sa se calculeze media. Afisarea sa se faca in urmatorul format:

```
coloana
0          10          15      20       25
|          |           |       |        |
v          v           v       v        v
numele     nota1       nota 2  nota3    media
```

Indicatie: utilizati 5 enunturi DATA sub forma: DATA "ANDREI";7,9,10 numele fiind de maximum 10 caractere.

3. Scrieti un program pentru numerotarea coloanelor si liniilor ecranului, sub forma

```
01234567890123456789012345678901
1
2
3
.
.
.
21
```

Raspunsuri

R 7.1 a)

```
20     16
12      8
 4      0
STOP
```

b)

```
224
235
246
426
437
448
628
```

```
        639
       6410
R 7.2 a) variabila nu poate fi o expresie
      b) nu se pot folosi variabile sir
      c) variabila de la FOR nu se poate pune la STEP
      d) variabile diferite la FOR si NEXT
R 7.3 a) FOR n = 2 TO 60 STEP 2

      b) 10 FOR x=1 TO 10
         20 IF x > 6 THEN GO TO 40
         30 NEXT x
         40 PRINT x: PRINT "abc"

      c) FOR n = 0.8 TO (a↑2 + 3) STEP (a+b)
```

CAPITOLUL 8
CUM FOLOSIM COLECTII DE DATE?

Pe masura ce dvs. deveniti un programator tot mai experimentat, incepeti sa aveti pretentia justificata sa cunoasteti si posibilitatile mai deosebite ale calculatorului la care lucrati.

Tablouri de numere/siruri

Una din aceste posibilitati este aceea de a folosi, intr-un mod relativ simplu, colectii mari de date.

Pentru o mai usoara intelegere a notiunilor, sa le comentam putin inainte de trece la aplicatia practica.

DIM. GOSUB. RETURN

Daca intr-un program se utilizeaza citeva numere, cinci de exemplu, ele pot fi identificate usor prin tot atitea variabile, sa zicem a, b, c, d, e. Cum vom proceda insa daca vrem sa lucram cu 100 de numere, sau siruri de caractere, diferite?

Tablou de date

Intr-o asemenea situatie se lucreaza cu un TABLOU ,DE DATE.

Un TABLOU reprezinta o colectie de date identificata printr-un singur nume de variabila.

Un tablou contine ELEMENTE si poate fi cu o singura dimensiune, sau poate avea doua dimensiuni.

Un tablou unidimensional se mai numeste LISTA de elemente.

Fie urmatorul sir de numere:

14, 8, 9, 11, 16, 20, 5, 3

Pentru a putea localiza oricare din aceste numere, le consideram intr-o lista cu opt elemente, careia ii dam un nume, sa zicem "a". Elementul 1 al listei contine valoarea 14, elementul 2 contine 8 s.a.m.d.

Numar element	1	2	3	4	5	6	7	8
Continut	14	8	9	11	16	20	5	3

a_1 (se citeste " a indice 1") contine "14",
a_4 contine "11", iar a_6 contine " ".

Intr-un tablou bidimensional, fiecare element poate fi localizat prin doua coordonate

	1	2	3	4	
1	3	-1	10	9	$b_{1,4}$
2	2	4	-2	6	$b_{2,4}$

Daca vom da tabloului numele "b", valorile cuprinse in el pot fi

identificate astfel:

$b_{1,3}$ = 10 -elementul din rindul 1, coloana 3
$b_{2,3}$ =-2 $b_{1,4}$ =9 $b_{2,1}$ =____
$b_{1,2}$ =___ $b_{2,4}$ =____

In BASIC indicii se scriu in paranteza:

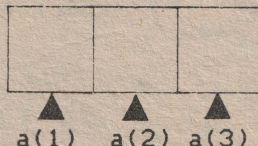
a_1 se scrie a(1) = 14
a_8 se scrie a(8) = 3
a(3) = ___
a(5) = ___
$b_{1,3}$ se scrie b(1,3) = 10
$b_{2,3}$ se scrie b(2,3) = -2
b(2,4) = ___
↑ ↑
rindul coloana

Aplicatia practica AP8

1. Introduceti:

```
10 DIM a(3)
20 LET a(1) = 6
30 LET a(2) = 33
40 LET a(3) = 999
50 PRINT a(1), a(2), a(3)
```

Notati rezultatele in: lista "a"



2. Adaugati linia:

```
60 PRINT : PRINT a(1) + a(2)
RUN
```

Ati obtinut afisarea continutului celor trei elemente ale listei si suma primelor doua.

3. Tastati:

```
50 FOR i=1 to 3
55 PRINT a(i),
60 NEXT i
RUN
```

4. Modificati linia 50 pentru a fi afisate numai primele 2 elemente.

```
50 --------------
RUN
```

5. Introduceti:

```
50 FOR i=1 TO 4
RUN
```

Notati mesajul obtinut ____________________________

Vi s-a semnalat ca ati cerut un al 4-lea element al unei liste declarata doar cu 3 elemente prin instructiunea:

```
DIM a(3)
```

6. Anulati programul existent si introduceti:

```
5 REM tablou cu 6 elemente
10 DIM b(2,3)
20 LET b(1,1) = 3
30 LET b(1,2) = -1
40 LET b(1,3) = 10
50 LET b(2,1) = 2
60 LET b(2,2) = 4.25
70 LET b(2,3) = -2
80 PRINT b(1,1),b(1,2),b(1,3),b(2,1),b(2,2),b(2,3)
RUN
```

Completati rezultatele in tabloul "b"

b(1,3)

b(2,1)

7. Schimbati programul astfel:

```
80 FOR i=1 TO 2          (rinduri)
90 FOR j=1 TO 3          (coloane)
100 PRINT b(i,j)
110 NEXT j
120 NEXT i
RUN
```

Notati rezultatele:

```
b(1,1)=____   b(1,2)=____   b(1,3)=____
b(2,1)=____   b(2,2)=____   b(2,3)=____
```

8. Schimbati programul:

```
NEW
10 DIM a$ (12)
20 LET a$ = "ABCDEFGHIJKL"
25 PRINT "123456789012"
27 PRINT "ABCDEFGHIJKL"
30 INPUT "numar element=";ne
40 PRINT a$(ne)
50 GO TO 30
RUN
```

Notati ce se obtine:

```
numar element = 1     ____
              = 4     ____
              = 9     ____
              =10     ____
              =11     ____
```

9. Anulati liniile 25, 27 si schimbati:

```
30 FOR n=1 TO 10
40 PRINT a$(n);
50 NEXT n
```

```
RUN
```

Obtineti continutul listei de caractere.

10. Introduceti:

```
NEW
10 DIM a$(5)
20 FOR n=1 TO 5
30 READ a$(n)
40 PRINT a$(n)
50 NEXT n
60 DATA "A","B","C","D","E"
RUN
```

11. Tastati:

```
5 REM tablou cu 5 rinduri si 2 coloane
10 DIM t(5,2)
15 REM r=rind, c=coloana
20 FOR r=1 TO 5
30 FOR c=1 TO 2
40 READ t(r,c)
50 NEXT c: NEXT r
60 REM afisarea tabloului
70 FOR r=1 TO 5
80 FOR c=1 TO 2
90 PRINT t(r,c),
100 NEXT c: NEXT r
110 DATA 1,2
120 DATA 3,4
130 DATA 5,6
140 DATA 7,8
150 DATA 9,10
RUN
```

Comparati ce s-a afisat, cu ce contin enunturile DATA.

12. Introduceti alt program:

```
NEW
10 REM program cheltuieli rechizite
15 REM ====================
20 REM afisare cap tabel
30 GO SUB 100
35 LET total = 0
40 REM =======================
45 REM introducere rechizite
50 GO SUB 150
60 REM =======================
65 REM calcul cheltuieli
70 GOSUB 210
80 GO TO 40
90 REM =====================
91 REM calcul total
95 GO SUB 300
99 STOP
100 PRINT "obiect";TAB9;"bucati";TAB18;"pret";TAB25;"cost"
110 PRINT "*******************************"
120 PRINT
130 RETURN
```

```
150 INPUT "obiect? ";o$
152 IF o$="gata" THEN GO TO 90
155 INPUT "bucati?␣";b,"pret unitar?␣";p
160 RETURN
210 LET cost = b*p
220 PRINT o$;TAB9;b;TAB18;p;TAB25;cost
230 LET total = total + cost
240 RETURN
300 PRINT "================================"
305 PRINT
310 PRINT "TOTAL:␣";TAB25;total
330 RETURN
RUN

obiect? "caiete"
bucati? 3          pret? 3.5

obiect? "creioane"
bucati? 12         pret? 0.9

obiect? "linii"
bucati? 15         pret? 2

obiect? "gata"
................rezultate.......
STOP
```

In acest program au aparut doua noi instructiuni BASIC pe care nu le-ati intilnit pina acum:

GO SUB - prescurtare de la GO SUBROUTINE (in traducere:salt la subrutina)
RETURN - (in traducere:intoarcere)

Aceste instructiuni permit o structurare a programelor astfel incit sa se evite scrierea repetata a anumitor parti (denumite subrutine, sau subprograme).

Analizati programul de mai sus.

El este structurat cu o parte principala cuprinsa intre prima linie (10) si linia STOP (99) si alte parti secundare denumite SUBRUTINE (subprograme).

In cadrul programului principal se fac trimiteri la subprogramele (subrutinele) pentru:

-afisarea capului de tabel
-introducerea rechizitelor cumparate
-calculul costului acelor rechizite (cheltuieli)
-calculul sumei totale cheltuite

O subrutina incepe cu linia data in GOSUB si se termina cu RETURN care produce intoarcere la programul principal. De exemplu, afisarea capului de tabel este facuta de subrutina continind liniile (priviti in program):

```
100 PRINT...
.
.
.
130 RETURN
```

Subrutina este apelata prin linia:

```
30 GO SUB 100
```

Urmeaza executia liniilor subrutinei pina la linia 130 care produce intoarcerea la linia care urmeaza dupa GOSUB adica:

35 LET...

Incercati sa urmariti singuri logica programului si completati informatiile lipsa:

Calculul totalului este determinat de linia:___ GOSUB ____. Subrutina incepe cu linia _____ si se termina cu linia 330 ______ dupa care se executa linia 99.

Introducerea rechizitelor cumparate, a cantitatii si a preturilor corespunzatoare se face ca urmare a liniei _____ GO SUB _______.

Subrutina incepe cu linia _____ si se termina cu linia _____ RETURN dupa care se executa linia _______ REM ______________________________.

Ati remarcat probabil ca pentru a termina executia introduceti cuvintul "gata" (adica s-au terminat cumparaturile), Obiectul? "gata". Se afiseaza rezultatele. STOP.

Notiuni de baza

O colectie de date poate fi memorata in programe BASIC, sub forma de TABLOU continind ELEMENTE.

Pentru a localiza orice element intr-un tablou, programatorul trebuie sa cunoasca numele tabloului si pozitia elementului in el. Aceasta pozitie este data prin indexarea (numerotarea) elementelor.

LISTA

Un tablou in care orice element este localizat printr-un singur indice se numeste LISTA.

Fie de exemplu o lista cu numele "l". Un element al listei este localizat prin:

```
              l(i)
              ↑ ↑
nume lista────┘ └──pozitia / numarul elementului in lista
```

MATRICE

Un tablou bidimensional mai este denumit si MATRICE si contine elemente ce pot fi localizate prin doi indici.

De exemplu, intr-o matrice numerica cu numele "t", un element este localizat prin rindul (r) si coloana (c) in care se afla:

```
                          t(r,c)
                          ↑ ↑ ↑__coloana
nume tablou/matrice───────┘ └────rind
```

Elementele unui tablou pot fi:
.numere reale-(numele tabloului este o singura litera);
.sir de caractere-(numele tabloului este o litera urmata de $)

DIM

Inainte de a fi utilizat un tablou, in program se pune o instructiune de rezervare a spatiului corespunzator in memorie.

Instructiunea DIM (de la DIMENSION - in engleza) declara tablouri de numere sau de siruri de caractere.

Sintaxa: DIM l(i) sau DIM l$(i) - pentru LISTA
DIM t(r,c) sau DIM t$ (r,c) - pentru MATRICE

```
l,t - nume ale tabloului
i   - numarul elementului in lista
r,c - numarul rindului, coloanei la intersectia carora se afla
      elementul
```

Cu o instructiune DIM se defineste numai un singur tablou:

```
10 DIM a(i), b(j)  - gresit
10 DIM a(i) : DIM b(j)  - corect
```

Instructiunea DIM are urmatoarele efecte:

1- rezerva pentru tablou spatiul corespunzator

```
DIM a(7)   - spatiu pentru 7 elemente
DIM t(3,2) - spatiu pentru 6 elemente (3*2)
```

2- initializeaza elementele tabloului cu zero

Incercati:

```
10 DIM a(3)
20 PRINT a(1);a(2);a(3)
RUN
```

3- sterge orice alt tablou cu acelasi nume

In program puteti da acelasi nume unui tablou si unei variabile simple, fara sa apara eroare deoarece numele tabloului este intotdeauna indexat. Este deci permis sa aveti:

```
LET a=...     variabila simpla
LET a(i)=...  tablou
```

Un tablou de siruri nu poate avea acelasi nume cu o variabila sir simpla.

Un tablou de siruri are un sir in fiecare rind, toate fiind de aceiasi lungime (completate sau scurtate pina la aceeasi lungime declarata):

```
        DIM g$(3,6)
               ▲ ▲
   3 siruri────┘ └──6 caractere in fiecare sir
```

Tastati:

```
10 DIM g$(3,6)
20 LET g$(1)="abcdef"
30 LET g$(2)="sirul2"
40 LET g$(3)="mdgip␣"
50 PRINT g$(1)'g$(2)'g$(3)
RUN
```

Obtineti continutul matricii:

```
abcdef  ◄──── rindul 1
sirul2  ◄──── rindul 2
mdgip␣  ◄──── rindul 3
▲_________6 caractere
```

Completati cu:

```
                              se
60 PRINT g$(1,4)          - se va afisa d
70 PRINT g$(2,4)          - se va afisa __
80 PRINT g$(3,3 TO 5)     - se va afisa gip
             ▼      ▼
       sirul 3   caracterele 3 la 5 (3,4 si 5)
```

Adaugati:

```
90 PRINT g$(1,1) + g$(2) + g$(3,1 TO 2)
```

Ce credeti ca va afisa linia 90? _____________________________________

Executati programul si verificati!

In liniile 80 si 90 au aparut doua lucruri noi:

1-se pot referi odata mai multe caractere ale unui sir sub forma:

```
                     m TO n
                     ↑    ↑
de la acest caracter─┘    └─pina la acest caracter (inclusiv)
```

2-se pot alipi caractere din diferite siruri prin localizarea si insumarea lor.

SUBRUTINE

Un program BASIC daca are anumite parti care se repeta, poate fi structurat astfel incit aceste parti sa fie scrise o singura data si sa fie apelate ori de cite ori este nevoie. Astfel de parti de program sint denumite SUBRUTINE.

GO SUB

Instructiunea GO SUB nr. linie produce saltul in program la subrutina care incepe cu linia indicata si se termina cu prima instructiune RETURN.

RETURN

Instructiunea RETURN este obligatorie pentru incheierea unei subrutine si provoaca intoarcerea executiei programului la prima instructiune dupa GO SUB.

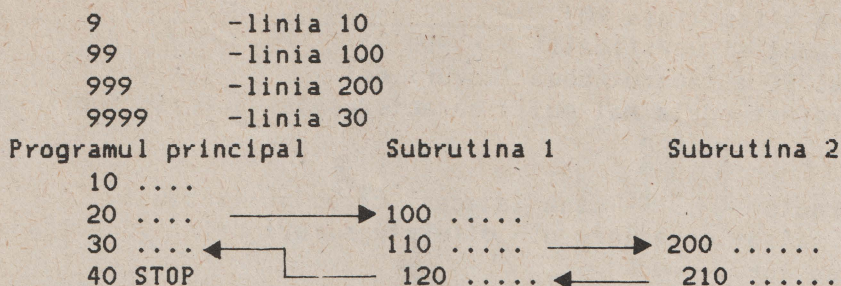
```
10...                 - inceput program principal
.
.
80 GO SUB 200         - salt la linia 200
90...
.
.
130 GO SUB 200         - salt la linia 200
140...
190 STOP               - sfirsit program principal
200 REM subrutina      - inceput subrutina
.
.
280 RETURN             - sfirsit subrutina
```

Linia 280 produce intoarcerea in programul principal la liniile 90 si 140.

In general subrutinele sint amplasate spre sfirsitul programului.

Este permis apelul unei subrutine dintr-o alta subrutina. De exemplu:

```
10 PRINT 9
20 GO SUB 100
30 PRINT 9999
40 STOP
100 PRINT 99
110 GO SUB 200
120 RETURN
200 PRINT 999
210 RETURN
RUN
```



Exemple de programare

Exemplul 1.

Program pentru declararea unei liste cu 9 elemente care sa contina valorile 1 2 3 4 5 6 7 8 9 si sa totalizeze suma dupa fiecare element introdus de la tastatura. Rezultatele sa fie afisate astfel:

continut element	suma	
1	1	
2	3	(2+1)
3	6	(3+3)
.	-	
9	-	

Variabile:

l - numele listei
n - numarul elementului din lista
s - suma elementelor

Codificarea programului:

```
 10 REM nume program: LISTA
 20 DIM l(9)
 30 LET s=0
 40 PRINT "introduceti in lista valorile:"
 50 PRINT "123456789"
 60 FOR n = 1 TO 9
 70 INPUT "n=";l(n)
 80 PRINT n, s+l(n)
 90 LET s=s+l(n)
100 NEXT n
110 STOP
```

Exemplul 2.

Program pentru insumarea valorilor dintr-un tablou bidimensional.

Analizarea problemei:

Fie un tablou cu un numar oarecare de linii si coloane dat la cerere.

Programul sa citeasca datele de calcul intr-unul sau mai multe enunturi DATA, memorindu-le in tabloul dimensionat corespunzator.

Sa se faca suma pe linii si coloane, precum si totalul general.

Pentru a intelege mai bine problema sa consideram, ca un exemplu particular, un tablou cu 4 linii si 5 coloane, care sa contina:

linii\coloane	1	2	3	4	5	TOTAL pe linii
1	1	2	3	4	5	15
2	6	7	8	9	10	40
3	11	12	13	14	15	65
4	16	17	18	19	20	90
TOTAL pe coloane	34	38	42	46	50	210

TOTAL GENERAL

Programul propus spre realizare trebuie deci sa calculeze:

- suma pe linie; de exemplu pe linia 1:
  1+2+3+4+5=15
- suma pe coloana, de exemplu pe coloana 1:
  1+6+11+16=34
- suma totalurilor liniilor:
  15+40+65+90=210
- suma totalurilor coloanelor:
  34+38+42+46+50=210

Bineinteles totalul general pe linii si pe coloane trebuie sa fie acelasi.

Afisarea rezultatelor sa se faca astfel:

1. tabloul cu date

```
    continutul primei linii
    .
    .
    .
    continutul ultimei linii
    .totalizarea pe linii
    linia 1        total:
    .
    .
    .
    linia l        total:
    total general=
```

2. tabloul cu date (vizualizat din nou pentru a verifica usor totalizarea verticala)

```
    . totalizarea pe verticala
    coloana 1      total:
    .
    .
    coloana c      total:
    total general=
    STOP
```

Variabile:

```
    nl - numar de linii
    nc - numar de coloane
    t(nl,nc) - tabloul de calculat
    s(nl) - lista unei linii
    w(nc) - lista unei coloane
    l - numar linie
```

c - numar coloana
s - suma totalurilor pe linie (total general pe orizotala)
s(l) - total unei linii
v - suma totalurilor pe coloana (total general pe verticala)
v(c) - totalul unei coloane
t(l,c) - tabloul cu date

Codificarea programului:

```
  5 REM cerere numar linii si coloane
  6 REM -----------------------------
 10 INPUT "nl=";nl,"nc=";nc
 15 REM declararea dimensiunilor tablourilor
 16 REM ----------------------------------
 20 DIM t(nl,nc):DIM s(nl):DIM v(nc)
 30 REM citirea unei linii
 35 REM -----------------
 40  FOR l=1 TO nl
 50   FOR c=1 TO nc
 60    READ t(l,c)
 70   NEXT c
 80  NEXT l
 90 REM afisarea datelor
 95 REM ---------------
100 GO SUB 600
110 REM totalizare orizontala
115 REM --------------------
120 PRINT
130 PRINT "totalizare pe linii"
140 PRINT
150 LET s=0
160  FOR l=1 TO nl
170   LET s(l) = 0
180    FOR c=1 TO nc
190     LET s(l) = s(l) + t(l,c)
200    NEXT c
210   PRINT "linia␣";l,"total:␣";s(l)
220   LET s=s+s(l)
230  NEXT l
235 PRINT
240 PRINT "total general=␣";s
250 PRINT
260 REM afisarea datelor
265 REM ---------------
270 GO SUB 600
280 REM totalizare verticala
285 REM -------------------
290 PRINT
300 PRINT "totalizare verticala"
310 PRINT
320 LET v=0
330  FOR c=1 TO nc
340   LET v(c) = 0
350    FOR l=1 TO nl
360     LET v(c) = v(c) + t(l,c)
370    NEXT l
380   PRINT "coloana␣";c,"total:␣";v(c)
390   LET v=v+v(c)
400  NEXT c
405 PRINT
```

```
410 PRINT "total general=␣";v
500 DATA 1,2,,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
550 STOP
600 REM rutina pentru afisarea datelor
605 REM ----------------------------
610 PRINT "tabloul cu date"
620 PRINT
630  FOR l=1 TO nl
640    FOR c=1 TO nc
650      PRINT t(l,c)
660    NEXT c
670   PRINT
680  NEXT l
690 RETURN
```

Analizati programul si sub aspectul scrierii lui (cu linii decalate) astfel incit sa fie cit mai usor de urmarit.

Atentie! Daca vi se afiseaza mesajul: E Out of DATA, 60:1 nu ati introdus in linia 500 date intr-un numar egal cu: numar linii * numar coloane.

Intrebari recapitulative si exercitii

I 8.1 Ce tipuri de tablouri (lista/matrice) sint urmatoarele si cite elemente contin?

a) DIM A$(20)
b) DIM q(5,10)
c) DIM w$(10,16)
d) DIM F(8)

I 8.2 Ce erori contin urmatoarele enunturi?

a) DIM a(6.8,14)
b) DIM b(k,-3)
c) DIM x(7), y(15)
d) DIM l

I 8.3 Scrieti instructiunile necesare pentru executarea urmatoarelor operatii (folosind cicluri FOR NEXT):

a) sa incarce cu "1" o matrice cu 10 linii si 10 coloane si sa se afiseze in format de 10*10
b) sa genereze, sa incarce si sa se tipareasca o lista cu urmatorul continut:

```
 2      4
 8     16
32     64
```

I 8.4 Fie o lista a(8). Sa se afiseze indicele (i) si valoarea (v) pentru toate elementele mai mici decit 15, cuprinse in:

```
DATA 1,12,17,4,28,20,2,13
```

Afisarea sa se faca incepind din coloanele 3 si 12.

```
coloana 3     coloana 12
   ↓              ↓
  i=1            v=1
```

I 8.5 Ce va afisa urmatorul program?

```
10 LET n=10
20 GOSUB 100
30 LET n=n/2
40 GO SUB 100
```

```
 50 LET n=n+1:GOSUB 100
 60 STOP
100 LET s=0
110 FOR i=1 TO n:LET s=s+i
120 NEXT i
130 PRINT s: RETURN
```

Probleme

1. Pentru un tablou numeric "a" cu x linii si x coloane, scrieti un program care sa calculeze produsul:

a(1,1)*a(2,2)*a(3,3)*...*a(x,x)

2. Program pentru afisarea elementelor coloanei a sasea a unui tablou T(8,10)

3. Program pentru afisarea liniei a treia a unui tablou d(5,8)

4. In instructiuni DATA puneti notele obtinute de 15 elevi la 5 materii. Realizati programul pentru calculul mediilor.

Tabloul datelor:

elevi → / materii ↓	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	10	9	9	7	6	10	5	6	10	10	8	7	9	8	10
2	9	7	8	6	6	8	6	7	9	10	9	7	9	8	9
3	9	5	7	7	5	8	7	8	9	10	10	8	10	8	10
4	8	6	6	8	7	10	6	6	10	9	8	8	10	9	9
5	10	8	10	7	5	9	5	7	9	8	9	8	9	9	10

Rezultatele sa fie afisate astfel:

```
elev        media
----        ------
 1            ?
 .            .
 .            .
 .            .
 15           ?
```

Raspunsuri

R 8.1 a) lista alfanumerica - 20 elemente
b) matrice numerica - 50 elemente
c) matrice alfanumerica - 160 elemente
d) lista numerica - 8 elemente

R 8.2 a) indice zecimal nu este permis
b) indice negativ nu este permis
c) cu o instructiune DIM se defineste un singur tablou; corect: DIM x(7):DIM y(15)
d) nume de lista fara indice

R 8.3 a)
```
10 DIM n(10,10)
20 FOR l=1 TO 10
30 FOR c=1 TO 10
40 LET n(l,c)=1
```

```
50 NEXT c
60 NEXT l
70 FOR l=1 TO 10
80 FOR c=1 TO 10
90 PRINT n(l,c);
100 NEXT c
110 PRINT
120 NEXT l
```

b)
```
10 DIM a(6)
20 LET a(1)=2:PRINT a(1),
30 FOR l=2 TO 6
40 LET a(l)=2*a(l-1)
50 PRINT a(l),
60 NEXT l
```

R 8.4
```
10 DIM a(8)
20 FOR i=1 TO 8
30   READ a(i)
40   IF a(i) > 15 THEN GO TO 60
50   PRINT TAB 3; "i=";i;TAB 12;"v=";a(i)
60 NEXT i
70 DATA 1,12,17,4,28,20,2,13
```

R 8.5
```
55
15
66
```

CAPITOLUL 9
SA DESENAM CU CIP-UL

In acest capitol veti cunoaste alte citeva posibilitati ale calculatorului cu care lucrati:

.moduri de afisare care sa puna in evidenta numai anumite informatii de pe ecran;

.realizarea de desene pe ecran

.obtinerea efectului de animatie

Instructiuni de utilizare a culorilor.
Instructiuni grafice. Numere aleatoare.
Caractere definite de utilizator

Cip-ul permite colorarea imaginilor, atit alfanumerice cit si grafice, astfel incit daca puteti utiliza un televizor/monitor color, lucrul cu calculatorul devine si mai fascinant.

Puteti folosi opt culori prin actionarea in anumite instructiuni a cifrelor de la 0 la 7 astfel:

0 - negru	3 - mov	6 - galben
1 - albastru	4 - verde	7 - alb
2 - rosu	5 - albastru deschis	

Pe ecranul unui televizor /monitor alb-negru aceste culori apar ca nuante de la negru la gri deschis.

Cele opt culori pot fi utilizate in urmatoarele instructiuni:

BORDER 0...7 - colorarea conturului (bordurii) ecranului
PAPER 0...7 - colorarea fondului (hirtiei)
INK 0...7 - colorarea caracterului/punctului afisat

Stabiliti legatura cu televizorul color si executati aplicatia practica, observind cu atentie ce se intimpla.

Aplicatia practica AP9

1. Introduceti urmatoarele instructiuni imediate (nenumerotate) si notati ce culori se obtin pe contur.

```
BORDER 1     __________
BORDER 2     __________
BORDER 3     __________
BORDER 4     __________
BORDER 5     __________
BORDER 6     __________
BORDER 7     __________
BORDER 0     __________
```

2. Tastati:

```
10 FOR b=0 TO 7 : BORDER b: PAUSE 50: NEXT b
RUN
```

3. Introduceti urmatoarea instructiune si veti obtine afisarea in mijlocul ecranului a unui mesaj clipitor, cu litere albe pe fond albastru:

```
NEW
PRINT AT 11,10;PAPER 1;INK 7;FLASH 1;"OPRITI BANDA!"
```

S-a afisat in linia ______ incepind cu coloana ______
Afisarea clipitoare se obtine cu instructiunea FLASH 1, iar efectul intermitent se anuleaza cu FLASH 0.

4. Stergeti ecranul cu CLS si introduceti instructiunile necesare pentru a se afisa clipitor, in ultimele doua linii ale ecranului un mesaj in rosu si verde.

```
CLS
PRINT AT 20,1;FLASH 1;INK 2;"Pentru continuare";
INK 4;"␣apasati orice tasta!";INK 0
```

"Cerneala" (INK) rosie si verde a fost data de instructiunile ________ si ______.

5. Tastati urmatoarea linie care va afisa in linia 3, incepind din coloana 3, pe fond (hirtie) galben si verde:

```
CLS
PRINT AT 3,3;"Ecranul are";PAPER 6;"␣22 linii";
PAPER 7;"␣si␣";PAPER 4 ;"32 coloane"
```

6. Urmatoarea linie va afisa cu litere albe pe fond negru:

```
CLS
PRINT INVERSE 1; AT 4,10;"ALB"
RUN
```

Comanda INVERSE 1 inverseaza culorile pentru PAPER si INK.
Reveniti la afisarea curenta (fara CLS):

```
PRINT INVERSE 0;AT 4,10; "NEGRU"
RUN
```

Am lasat intentionat litera "B" afisata invers pentru a sesiza diferenta.

7. Introduceti:

```
CLS
PRINT AT 4,10 :FLASH 1:"clipitor:";FLASH 0;"/NORMAL"
```

8. In exemplul urmator realizati si afisarea color a unor linii, dar retineti si semnificatia textului care arata coordonatele extreme ale ecranului in afisarea grafica;

```
CLS
PRINT PAPER 6;AT 3,3;"punctele extreme sint:";PAPER 4;'␣ ';"
(0,0)␣␣␣␣␣␣-coltul  stinga  jos";'␣ ';"(0,175)␣␣␣- coltul
stinga sus";'␣ ';"(255,0)␣␣␣-coltul dreapta jos";'␣ ';
"(255,175)␣-coltul  dreapta sus"
```

Din aceasta linie lunga de program retineti doua aspecte noi:

- culoarea verde (PAPER 4) a fost scrisa o singura data, in continuare ea raminind stabilita si in liniile care au urmat;
- un blank intre apostrofuri ('␣') introduce o linie vida.

9. Introduceti urmatorul program, urmarind cu atentie comentariile:

```
CLS
5 REM marcarea punctelor extreme
10 PLOT 0,0 - coltul stinga jos
20 PLOT 255,0 - coltul dreapta jos (x max)
30 PLOT 0,175 - coltul stinga sus (y max)
```

```
40 PLOT 255,175 - coltul dreapta sus
RUN
```

Priviti cu atentie ecranul si veti observa cele patru puncte.

10. Adaugati liniile:

```
45 REM trasarea axei x
50 FOR x=0 TO 255: PLOT x,0 : NEXT x
55 REM trasarea axei y
60 FOR y=0 TO 175 : PLOT 0,y : NEXT y
RUN
```

11. Continuati introducerea:

```
 65 REM punct de coordonate x=150, y=100
 70 PLOt 150, 100
 75 REM trasarea abscisei punctului
 80 PLOT 0,100
 90 FOR x=0 TO 150 : PLOT x,100 : NEXT x
 95 REM trasarea ordonatei punctului
100 FOR y=0 TO 100 : PLOT 150,y : NEXT y
RUN
```

12. Anulati liniile 65,70,75,80,90,95,100 si modificati liniile:

```
50 PLOT 0,0 : DRAW 255,0
RUN
```

Se traseaza axa x dintr-o data si mai rapid decit prin puncte.

```
60 PLOT 0,0 : DRAW 0,175
RUN
```

Se traseaza rapid axa y.

```
65 REM desenarea diagonalei ecranului
70 PLOT 255,175
80 DRAW -255, -175
```

13. Desenati un patrat cu latura de 30.

```
NEW
5 BORDER 2
10 PLOT 50,50  - punctul de plecare
20 DRAW 30,0
25 PAUSE 50
30 DRAW 0,30
35 PAUSE 50
40 DRAW -30,0
45 PAUSE 50
50 DRAW 0, -30
RUN     <ENT>  <ENT>    (tastati de doua ori)
```

Instructiunea PAUSE am introdus-o pentru a observa treptat trasarea laturilor patratelor.

Anulati liniile 25, 35, si 45 si executati din nou programul.

Completati linia 5

```
5 BORDER 1: PAPER 5 : INK 7
```

```
RUN <ENT>
```

Ce s-a schimbat in privinta culorilor?

14. Descrieti un cerc cu raza de 30 si cu centrul in punctul de coordonate x=170 si y=120.

```
55 PLOT 170,120
60 CIRCLE 170,120,30
RUN
```

15. Desenati acum un cerc prin doua arce de cerc (semicercuri):

```
70 PLOT 40,120
80 DRAW 20,20,PI:PAUSE 50
90 DRAW -20,-20,PI
RUN
```

16. Sa recapitulam (vezi figura 5.1)

-punctul de origine la afisarea alfanumerica pentru instructiunea

```
PRINT AT linie, coloana
este in coltul  ________ = ______
```

-punctul de origine la afisarea grafica, pentru instructiunile

```
PLOT x, y
DRAW x, y
CIRCLE x, y, r
este in coltul ______ _ _______.
```

17. Introduceti un nou program:

```
NEW
 5 BORDER 6
10 FOR n=1 TO 10
20 LET x = 100 * RND
30 LET y = 170 * RND
40 PLOT 120, 0
50 DRAW x, y
60 PLOT 120, 0
70 DRAW -x, y
80 NEXT n
RUN
```

Iata o noua surpriza:

CIP-ul poate obtine efecte neasteptate (deosebit de utile la jocuri), generind numere la intimplare, intre anumie limite date prin program. Acestor numere li se mai spune numere ALEATOARE (in engleza RANDOM = aleator, la intimplare).

Linia 20 va produce la intimplare numere cuprinse intre 0 si 100 pentru x, iar linia 30 va produce numere aleatoare intre 0 si 170 pentru y.

18. Schimbati in program:

```
5 BORDER 4: FOR i=6 TO 2 STEP -1
50 DRAW INK i ; x, y
70 DRAW INK i; -x, y
90 NEXT i
RUN
```

19. Inca o schimbare in program:

```
50 DRAW PAPER i-1; INK i; x, y
70 DRAW PAPER i-1; INK i; -x, y
RUN
```

Observati ca instructiunea DRAW permite utilizarea optiunilor de culoare pentru PAPER si INK, in mod asemanator cu PRINT. Executati de citeva ori programul cu RUN.

20. Sa analizam acum putin numerele aleatoare generate prin instructiunea RND, reluind primele linii din programul anterior:

```
 NEW
10 FOR n=1 TO 4
20 LET x = 100 * RND
30 LET y = 170 * RND
40 PRINT x,y
50 NEXT n
RUN
```

Notati valorile afisate:

```
---------       -------
---------       -------
---------       -------
---------       -------
```

Executati din nou programul.

```
RUN
```

Comparati cele afisate acum cu cele inscrise mai sus si observati valorile diferite pentru x si y.

Daca doriti numere intregi aleatoare schimbati de exemplu linia 30:

```
30 LET y = INT (170 * RND)
RUN
```

Notati:

```
----------     -------
----------     -------
----------     -------
----------     -------
```

De data aceasta y ia valori intregi

21. Si acum un program pentru trasarea graficului functiei trigonometrice SIN intre 0 si 2π cu doua culori si doua valori de amplitudine:

```
NEW
10 BORDER 6
20 PLOT 0,88 : DRAW 255,0
30 FOR y = 0 TO 255
40 PLOT INK 4; y, 88 + 40 * SIN (y/128 * PI)
50 PLOT INK 1; y, 88 + 85 * SIN (y/128 * PI)
60 NEXT y
RUN
```

22. Introduceti:

```
NEW
1 REM folosirea caracterelor grafice standard█si▀
```

```
  5 BORDER 6
 10 PRINT AT 18, 6 ; INK 2 ; "████████████████"
 20 LET l=17
 30 LET y$ = "▄▀"
 40 FOR c=6 TO 20
 50 PRINT AT l, c; y$
 60 LET l = l -1
 70 NEXT c
 80 FOR l = 3 TO 18
 90 PRINT AT l,21;INK 2;"█"
100 NEXT l
RUN
```

Obtineti o scara solida . Sa-i dam o utilizare facind sa sara pe ea o ... minge reprezentata prin litera "O"!

Veti invata astfel cum se realizeaza animatia pe ecran. Adaugati liniile:

```
110 FOR k=1 TO 5
120 LET lm=1
130 FOR c=21 TO 4 STEP -1
135 REM se deseneaza mingea
140 PRINT AT lm, c; INK 1; "O"
150 PAUSE 8
155 REM se sterge mingea
160 PRINT AT lm, c; "␣"
170 LET lm = lm + 1
180 NEXT c
190 REM mingea pe orizontala
200 FOR c=5 TO 22
210 PRINT AT 19,c ; INK 1; "O"
220 PAUSE 4
230 PRINT AT 19, c; "␣"
240 NEXT c
250 REM mingea pe verticala
260 FOR l=18 TO 1 STEP -1
270 PRINT AT l,22; INK 1; "O"
280 PAUSE 2
290 PRINT AT l, 22 ; "␣"
300 NEXT l
310 NEXT k
RUN
```

Animatia se obtine intr-un cicul FOR...NEXT, prin afisarea intr-o pozitie, mentinerea imaginii pe o anumita durata si stergerea imaginii, urmata de afisarea in pozitia urmatoare.

De exemplu ridicarea verticala a mingii este realizata cu liniile:

```
260 FOR l=18 TO 1 STEP -1  -scade numarul liniei
270 PRINT AT l, 22; "O"    -afiseaza "O"
280 PAUSE 2                -mentine imaginea
290 PRINT AT l, 22; "␣"    -sterge pe "O"
300 NEXT l                 -pozitia urmatoare
```

Notiuni de baza

Daca aveti la dispozitie un televizor color, cu ajutorul CIP-ului puteti realiza opt culori numerotate astfel:

1: albastru 2: rosu 3: magenta

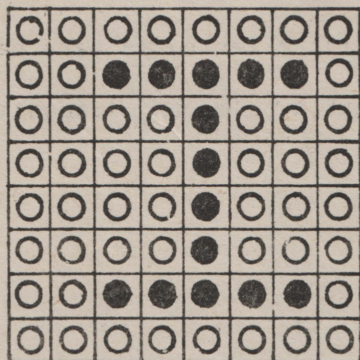
4: verde 5: bleu (cyan) 6: galben

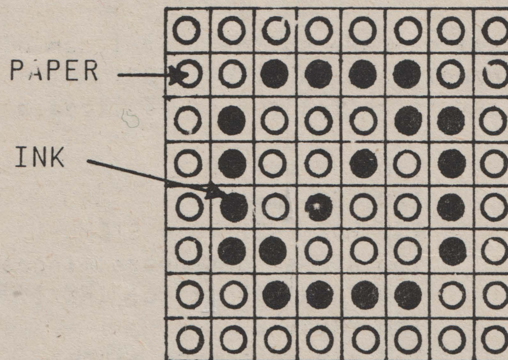
7: alb 0: negru

Un caracter este afisat intr-un patrat format din 8*8 puncte. De exemplu litera "I" si cifra "O" sint afisate astfel:

INK PAPER

Pentru fiecare caracter se pot folosi doua culori:
INK c - culoarea punctelor negre din desen
PAPER c - culoarea punctelor albe din desen, c putind avea valoarea 0 la 7





BORDER

Instructiunea BORDER c coloreaza marginea imaginii cu culoarea data de valoarea lui c.
BORDER 2 - margine rosie
BORDER 6 - margine galbena
Culorile si afisarea clipitoare (FLASH) sint considerate ca ATRIBUTE ce pot fi asociate unui caracter.
Sa afisam litera I cu cerneala (INK) albastra pe fond (PAPER) galben:

```
NEW
10 FOR n=1 TO 5
20 PRINT PAPER 6; INK 1; "I";
30 PRINT PAPER 7; INK 7; "␣";
40 NEXT n
```

Atributele de culoare pot fi utilizate fie separate, fie in instructiunea PRINT.
Urmatorul program va afisa cele opt culori si cifrele corespunzatoare:

```
NEW
10 FOR n=1 TO 80
20 FOR c=0 TO 7
30 PAPER c
40 PRINT c;
50 NEXT c: NEXT n
RUN
```

Daca dorim ca cifrele sa fie mai evidente, se poate obtine un contrast mai bun introducind

INK 9 sau PAPER 9

Cifra 9 nu este asociata unei culori, ci are ca efect cerneala mai

deschisa pe fond inchis (negru, albastru, rosu, magenta) si cerneala mai inchisa pe fond deschis (verde, bleu, galben, alb).

Verificati introducind linia:

```
15 INK 9
RUN
```

Atributele de culoare si clipire le puteti folosi cu mare efect, atunci cind doriti sa scoateti in evidenta anumite informatii de pe ecran.

Daca, de exmplu aveti ecranul plin si afisarea este uniforma, cel care analizeaza cele afisate trebuie sa parcurga tot continutul pentru a gasi o informatie. Afisind insa in alta culoare si eventual clipitor, imediat informatia respectiva atrage atentia privitorului.

Pozitionarea pe ecran se face cu

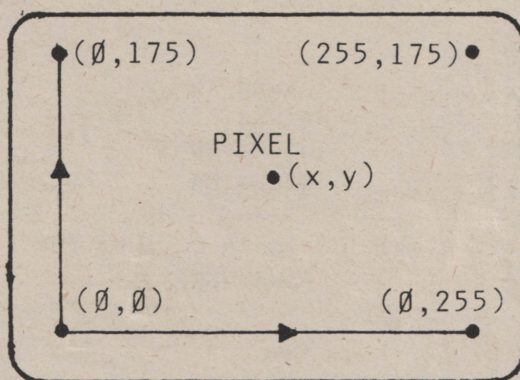
```
PRINT AT linie, coloana        unde: linie = 0 - 21
                                     coloana = 0 - 31
```

Atributele pot fi plasate in linia de program fie imediat dupa PRINT, fie dupa AT linie, coloana:

```
PRINT AT 11,10;PAPER 1;INK 7;FLASH 1;"OPRITI BANDA"
PRINT PAPER 1;INK 7;FLASH 1;AT 11,10;"OPRITI BANDA"
```

Realizarea desenelor pe ecran se obtine prin puncte.

Un punct este denumit PIXEL si poate fi pozitionat prin program, cu doua coordonate fata de coltul din stinga - jos al ecranului (vezi si figura 5.1)



PLOT

Instructiunea PLOT x, y stabileste pozitia si deseneaza un punct la coordonatele x, y:

x = 0 - 255
y = 0 - 175

Desenele pot fi afisate punct cu punct. De exemplu trasarea a doua axe la extremitatile ecranului se poate face astfel:

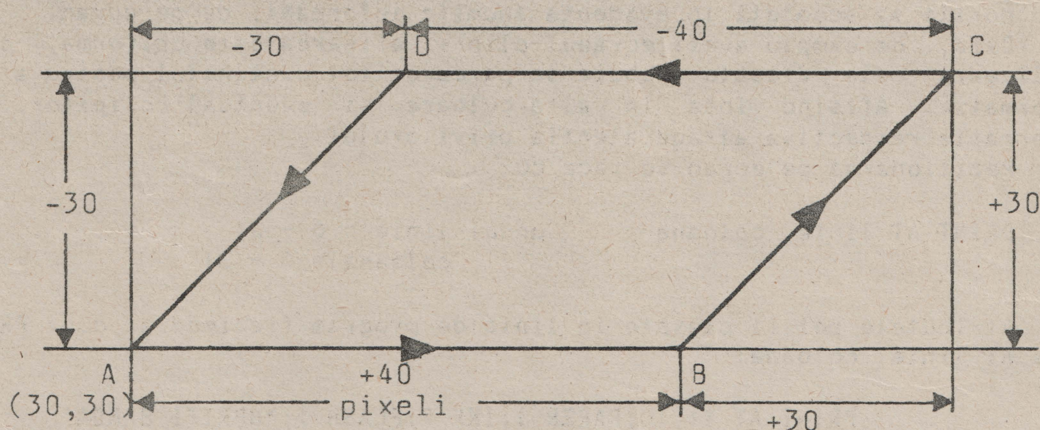
```
.axa x:   FOR x=0 TO 255 : PLOT x,0: NEXT x
.axa y:   FOR y=0 TO 175 : PLOT 0,y: NEXT y
```

DRAW

Instructiunea DRAW x,y deseneaza o linie de la pozitia curenta a cursorului grafic pina la un punct de coordonate x si y pixeli. De exemplu trasarea celor doua axe, luind ca punct de origine (10,10) se poate obtine cu:

```
NEW
10 PLOT 10, 10
20 DRAW 245, 0
30 PLOT 10,10
40 DRAW 0, 165
```

Pentru a intelege mai bine folosirea instructiunii DRAW sa analizam desenarea paralelogramului din figura:



Ne propunem sa incepem desenul din coltul A de coordonate x = 30 ; y = 30 si sa trasam laturile in sensul aratat.

```
NEW
10 PLOT 30, 30
20 DRAW 40, 0        - latura AB
30 DRAW 30, 30       - latura BC
40 DRAW -40, 0       - latura CD
50 DRAW -30, -30     - latura DA
```

Folosind un asemenea paralelogram in cicluri FOR...NEXT puteti obtine obti constructii spectaculoase ca in programul urmator:

```
NEW
10 BORDER 6
20 FOR n=1 TO 100 STEP 1.2
30 PLOT 30, 30 + n
40 DRAW 40,n-n
50 DRAW 30,(n-n)+30
55 BEEP 0.1, n/10
60 NEXT n
70 DRAW -40,n-n
80 DRAW -30,(n-n)-30
90 FOR n = 1 TO 105 STEP 2
100 PLOT 150, 50 + n
110 INK 2
120 DRAW 10,n-n
130 DRAW 15,(n-n)+15
135 BEEP 0.01, n/5
140 NEXT n
150 DRAW -10,n-n
160 DRAW -15,(n-n)-15
170 STOP
RUN
```

O alta forma a instructiunuii DRAW este:

DRAW x, y, z

z este unghiul (in radiani) pentru desenarea unei linii curbe intre pozitia cursorului grafic si pozitia data de x, y:

```
NEW
10 FOR i=1 TO 7: READ z
20 PLOT 125, 0
30 DRAW 0, 100, z
35 PAUSE 40: NEXT i
40 DATA PI/2, PI/4, 2*PI
50 DATA -PI/2, -PI/3, PI, -PI
```

Observati ca intre cele doua extremitati date de (125, 0) si (0, 100) se obtin curbe diferite in functie de valoarea lui z data in radiani.

CIRCLE

Instructiunea CIRCLE x, y, r deseneaza un cerc de raza r cu centrul in punctul de coordonate x si y

```
NEW
10 FOR i = 1 TO 4
20 READ raza
30 CIRCLE 100, 85, raza
40 DATA 20, 30, 50, 80
50 NEXT i
60 PLOT 100, 85
RUN
```

Sau:

```
NEW
10 BORDER 5
20 FOR r = 1 TO 50, STEP 2
30 CIRCLE INK 1; 127, 85, r
40 NEXT r
RUN
```

Generarea unor numere la intimplare

Adesea in programe (in special in jocuri) se folosesc numere diferite generate "la intimplare". Astfel de numere se numesc numere ALEATOARE (in engleza : RANDOM NUMBERS).

Imaginati-va diferite combinatii de numere date de aruncarea a doua zaruri si veti realiza ce se intelege prin numere aleatoare.

Exista diferite metode matematice pentru a produce numere aleatoare. CIP-ul genereaza astfel de numere cu functia RND.

RND

.RND - genereaza numere zecimale intre 0 si 0,99999999

```
NEW
10 FOR i=1 TO 20
20 LET n=RND
30 PRINT n,
40 NEXT i
```

. a * RND - genereaza numere zecimale intre 0 si a.
Schimbati linia 20:

```
20 LET n=15*RND
RUN
```

Executati programul de citeva ori si veti obtine numere intre 0 si 15.
.b + a*RND - genereaza numere zecimale intre b si (a+b)
Schimbati:

```
20 LET n = 2 + 15 * RND
```

Se obtin numere zecimale cuprinse intre 2 si 17.

```
20 LET n = 4.4 + 0.6 * RND
```

Se obtin numere intre 4,4 si 5

.INT (a*RND) -genereaza numere intregi intre 0 si a

```
20 LET n = INT(6*RND)
```

.b + INT(a+RND) -genereaza numere intregi intre b si (a+b)

```
20 LET n = 1 + INT(6*RND)
RUN
```

Executati de citeva ori programul si veti obtine numere aleatoare cuprinse intre 1 si 6, deoarece functia INT intregeste numarul la valoarea intreaga inferioara.

```
INT (3,65) = 3
INT (5,999) = 5
```

RAND (RANDOMIZE)

Uneori este insa nevoie de numere aleatoare in seturi care sa se repete. Aceasta posibilitate o da enuntul RANDOMIZE (prescurtat: RAND).
Pe tastatura obtineti (vezi anexa B):

"RAND" cu tasta T in modul [K]
"RND" cu tasta T in modul [E]

Pentru a vedea diferenta intre functia RND si enuntul RAND, introduceti urmatorul program:

```
NEW
50 FOR r = 1 TO 3
100 FOR i = 1 TO 5
110 LET a = 1 + INT(10 * RND)
120 PRINT a
130 NEXT i: PRINT: NEXT r
```

La fiecare executie a programului veti obtine trei seturi diferite de numere intregi aleatoare intre 1 si 10.
Completati programul cu liniile:

```
10 FOR x = 1 TO 5
20 INPUT "n=␣";n
```

```
 30 CLS
 70 RANDOMIZE n
140 NEXT n
```

Enuntul RAND are sintaxa: RAND n unde n = 0 la 65535.

Executati programul dind lui x valorile urmatoare si notati rezultatele pentru a sesiza diferentele si asemanarile intre ele:

```
n = 1      <ENT>
n = 70     <ENT>
n = 999    <ENT>
n=12345    <ENT>
n=65535    <ENT>
```

Analizind rezultatele notate, observati ca pentru o valoare data lui n se obtin aceleasi seturi de numere aleatoare, dar difera numerele din seturi, pentru diferite valori ale lui n.

RAND si RAND 0 utilizeaza timpul trecut de la punerea in functiune a calculatorului, care daca nu creste mult intre doua executii ale lui RANDOMIZE, determina generarea unor seturi cu numere aproximativ aceleasi.

Incercati executia programului repetind n=0!

Caractere grafice standard

Folosind caracterul grafic "█" , urmatorul program va va da un ecran clipitor:

```
NEW
10 BORDER 3
20 LET n = 1 + INT(7*RND): INK n
30 LET l = 1 + INT(21*RND)
40 LET c = 1 + INT(31*RND)
50 PRINT AT l, c; "█"
60 GO TO 10
```

Intrerupeti cind doriti, cu BREAK.

Caracterele grafice standard se obtin in modul [G] (CS+9), cu tastele numerice 1 la 8 (vezi anexa B). Ele sint utile cind doriti sa afisati texte sau grafice de dimensiuni mai mari decit cele obisnuite.

Introduceti:

```
 10 PRINT AT 8, 10 "□██□□□█□□███□"
 20 PRINT AT 9, 10 "█□□█□□█□□█□□█"
 30 PRINT AT 10,10 "█□□□□□█□□█□□█"
 40 PRINT AT 11,10 "█□□□□□█□□█□□█"
 50 PRINT AT 12,10 "█□□□□□█□□███□"
 60 PRINT AT 13,10 "█□□█□□█□□█□□□"
 70 PRINT AT 14,10 "□██□□□█□□█□□□"
 90 REM: trasare coloane
100 PLOT 0,0: DRAW 0, 175
110 FOR x = 7 TO 255 STEP 8
120 PLOT x,0 : DRAW 0,175
130 BEEP .001,20
140 NEXT x
190 REM:trasare linii
200 PLOT 0,0: DRAW 255,0
210 FOR y = 7 TO 175 STEP 8
220 PLOT 0, y : DRAW 255,0
230 BEEP .001,20
240 NEXT y
```

Cum se pot realiza caractere grafice speciale?

Asa cum am aratat, un caracter este reprezentat (afisat) intr-un patrat de 8 * 8 puncte. Folosind acest mod de reprezentare utilizatorul poate sa-si defineasca, simboluri grafice proprii introducind 0 pentru PAPER si 1 pentru INK.

Daca priviti in Anexa 1, codurile 144 la 164 sint rezervate pentru caractere grafice definite de utilizator (in engleza UDG-USER DEFINED GRAPHICS).

Fiecare UDG poate fi asociat prin program unei taste alfabetice, astfel incit sa poata fi obtinut oricind se trece in modul grafic si se actioneaza tasta respectiva.

Pentru exemplificare sa incercam sa definim litera greceasca "α" pe tasta <A>:

1. punctam noul caracter in 8 * 8 puncte lasind insa la margini cite un sir nefolosit pentru a separa caracterul de altele alaturate:

	1	2	3	4	5	6	7	8
1	O	O	O	O	O	O	O	O
2	O	O	O	O	●	●	O	O
3	O	O	O	●	O	O	●	O
4	O	O	O	●	O	O	O	O
5	O	O	●	●	O	O	O	O
6	O	●	O	●	O	O	●	O
7	O	O	●	O	●	●	O	O
8	O	O	O	O	O	O	O	O

2. se memoreaza fiecare din cele 8 rinduri cu enuntul BIN (provenit din BINAR) urmat de 8 cifre binare:
   - 0 - pentru fond hirtie (PAPER)
   - 1 - pentru cerneala (INK)

Cele opt numere binare rezulatate sint inscrise in memoria interna a CIP-ului in opt pozitii, fiecare pozitie avind o ADRESA. Adresa primului rind este USR"A" (USR de la USER si "A" de la tasta pe care am stabilit-o pentru noul caracter).

Patratul de 8 * 8 puncte devine:

¦OCTETI¦		adresele de memorare
BIN 0 0 0 0 0 0 0 0	▶	USR "A"
BIN 0 0 0 0 1 1 0 0	▶	USR "A" + 1
BIN 0 0 0 1 0 0 1 0	▶	USR "A" + 2
BIN 0 0 0 1 0 0 0 0	▶	USR "A" + 3
BIN 0 0 1 1 0 0 0 0	▶	USR "A" + 4
BIN 0 1 0 1 0 0 1 0	▶	USR "A" + 5
BIN 0 0 1 0 1 1 0 0	▶	USR "A" + 6
BIN 0 0 0 0 0 0 0 0	▶	USR "A" + 7

Numarul binar dat de un rind, fiind format din 8 biti (cifre binare) poarta numele de octet (in engleza BYTE)

Recapitulind putin:

adresa	octetul memorat
USR "A" + 2	0 0 0 1 0 0 1 0
USR "A" + 5	0 1 0 1 0 0 1 0
USR "A" + 7	0 0 0 0 0 0 0 0

Memorarea directa a unui numar la o anumita adresa se poate face cu

```
POKE adresa, numar
```

Urmatorul program va memora cei 8 octeti reprezentind caracterul "ℒ".

```
NEW
10 FOR r = 0 TO 7
20 READ b
30 POKE USR "a" + r, b
40 NEXT r
110 DATA BIN 0 0 0 0 0 0 0 0
120 DATA BIN 0 0 0 0 1 1 0 0
130 DATA BIN 0 0 0 1 0 0 1 0
140 DATA BIN 0 0 0 1 0 0 0 0
150 DATA BIN 0 0 1 1 0 0 0 0
160 DATA BIN 0 1 0 1 0 0 1 0
170 DATA BIN 0 0 1 0 1 1 0 0
180 DATA BIN 0 0 0 0 0 0 0 0
RUN
```

Dupa executia programului vi se afiseaza mesajul:

```
OK, 180:1
```

3.Introduceti linia:

```
200 PRINT "aaaa        (fara <ENT>)
```

4. Intrati in mod grafic si apasati din nou "A". Acum se afiseaza "ℒ". Completati linia 200:

```
200 PRINT "aaaa␣ℒℒℒℒ"
```

Atentie! Pentru a introduce ghilimelele de la sfirsitul sirului, reveniti din modul [G], in modul [L] reapasind (CS + 9)

```
RUN 200
```

Se afiseaza aaaa ℒℒℒℒ
De cite ori stabiliti modul [G], prin apasarea tastei <A> veti obtine "ℒ".

Animatia (miscarea) pe ecran

Efectul de miscare se obtine prin afisarea unui desen intr-o pozitie, mentinerea imaginii pe o durata scurta, stergerea desenului si reafisarea lui in pozitia urmatoare.

Mentinerea limitata a imaginii se realizeaza cu comanda:

```
PAUSE n
```

Opreste executia programului pe o durata data de valoarea lui n.

n = 0 - 65535

Pauza maxima este de aproximativ 22 minute.

```
PAUSE 65535
```

O pauza de cca 1 secunda va fi data de PAUSE 50.

Puteti opri programul pina la apasarea oricarei taste cu:

```
PAUSE 0
```

Programul urmator simuleaza saritura unei mingi lansata de o paleta. Vom folosi pentru minge litera "O" iar pentru paleta, semnul "\":

```
NEW
5 BORDER 1
10 REM mingea va fi lansata de 5 ori
20 FOR k = 1 TO 5
30 REM ridicarea mingii
40 FOR n = 21 TO 10 STEP -1
50 REM afisarea paletei
60 PRINT AT 21, 10; "\"
70 PRINT AT n, 10; "O"
80 PAUSE 5
90 PRINT AT n, 10; "␣"
100 NEXT n
110 REM caderea mingii
120 FOR n = 10 TO 21
130 PRINT AT n, 10; "O"
140 PAUSE 5
150 PRINT AT n, 10; "␣"
160 NEXT n
170 NEXT k
```

Exemple de programe

Sa realizam un program pentru a contoriza numarul de aparitii ale cifrelor 1 la 7 (utilizate pentru culori) generate de functia RND, dintr-un numar mare de executii.

Daca de exemplu cerem prin program generarea a 700 numere intregi aleatoare, intre 1 si 7, ne-am astepta ca sa obtinem fiecare cifra de 100 de ori. Este un mod de a vedea cum lucreaza generatorul de numere aleatoare cu care este dotat calculatorul dvs.

- Analiza problemei:

Se va genera un numar mare de numere aleatoare intregi (de ex. 700) intre 1 si 7 si vor fi contorizate (insumate) aparitiile fiecareia dintre ele.

Rezultatele sa fie afisate sub forma:

numar	de cite ori
1	140
2	96
.	
.	
.	
7	102

Pentru ca exemplul sa cuprinda cit mai multe din notiunile acestui capitol si pentru ca afisarea rezultatelor sa fie cit mai expresiva, sa se asocieze cifrelor 1 la 7 culorile corespunzatoare si sa se obtina sub liniile cu rezultate, un grafic vertical cu sapte coloane colorate a caror inaltime sa fie proportionala cu numarul de aparitii pentru fiecare cifra (culoare).

Pentru a insuma numarul de aparitii sa se foloseasca un tablou unidimensional cu 7 elemente declarat astfel:

```
10 DIM c(7)
```

- Codificarea programului:

```
 10 DIM c(7)
 15 PRINT FLASH 1; AT 10, 0; "ASTEPTATI cca 20 sec.- calculez!";
    FLASH 0
 18 REM generarea numerelor
 20 FOR k = 1 TO 700
 30 LET a = 1 + INT(7*RND)
 35 REM contorizarea aparitiilor
 40 LET c(a) = c(a) + 1
 50 NEXT k
 55 REM afisare rezultat
 60 CLS: PRINT "numar", "de cite ori"
 70 PRINT
 80 FOR p = 1 TO 7
 90 PRINT PAPER p; INK 9; p,c(p)
100 NEXT p
110 REM graficul rezultatelor
120 BORDER 0
130 PLOT 155,0:DRAW PAPER 1; INK 9;0,c(1)
135 BEEP c(1)/100,c(1)
140 PLOT 163,0:DRAW PAPER 2; INK 9;0,c(2)
145 BEEP c(2)/100,c(2)
150 PLOT 171,0:DRAW PAPER 3; INK 9; 0,c(3)
155 BEEP c(3)/100, c(3)
160 PLOT 179,0: DRAW PAPER 4; INK 9; 0, c(4)
165 BEEP c(4)/100, c(4)
170 PLOT 187, 0:DRAW PAPER 5; INK9; 0,c(5)
175 BEEP c(5)/100 , c(5)
180 PLOT 195,0:DRAW PAPER 6; INK9; 0,c(6)
185 BEEP c(6)/100, c(6)
190 PLOT 203,0 : DRAW PAPER 7; INK 9; 0,c(7)
195 BEEP c(7)/100,c(7)
200 STOP
```

Va propun sa incercati intelegerea programului raspunzind la intrebarile urmatoare:

Intrebari recapitulative

I 9.1 Ce instructiune a determinat:
- a) colorarea conturului ecranului?
- b) afisarea clipitoare a mesajului "ASTEPTATI..." ?
- c) Colorarea rezultatelor orizontale?

I 9.2 Care a fost efectul instructiunii INK 9?

I 9.3 Ce linie de program a trasat graficul vertical pentru aparitiile cifrei 3?

I 9.4 In ce linii de program trebuie sa faceti modificari pentru a genera 10000 de numere intregi aleatoare?

I 9.5 Ce instructiune banuiti ca genereaza sunete proportionale ca durata si inaltime cu numarul de aparitii ale cifrelor? (cautati singura instructiune neexplicata pina la acest capitol).

Probleme

Realizati programe pentru:

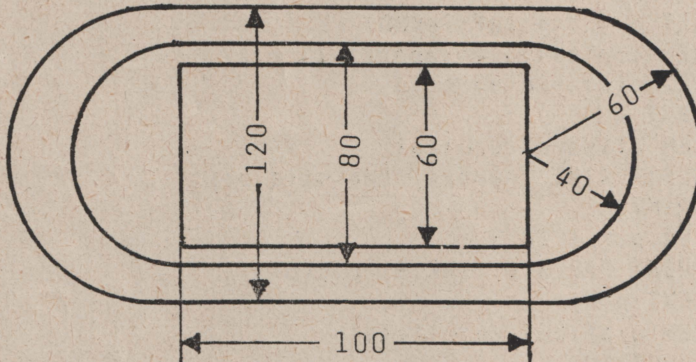
a) generarea si afisarea a 20 de numere aleatoare intre 0 si 100.
b) trasarea axelor x si y de lungime maxima, incepind din coltul din

dreapta sus.

c) umplerea ecranului cu puncte luminoase aleatoare, pe fond inchis (ca un cer instelat)

d) saritura amortizata a unei mingi (inaltimea sa descreasca pina la zero)

e) desenarea unui "stadion" de forma:



f) obtinerea literei grecesti β prin apasarea tastei "B".

Raspunsuri

R 9.1 a) 120 BORDER 0
b) 15 PRINT FLASH 1; AT 10, 0;
"ASTEPTATI..."; FLASH 0
c) 90 PRINT PAPER p...

R 9.2 Afisarea contrasta intre INK si PAPER

R 9.3 Linia 150

R 9.4 In linia 15 pentru a modifica timpul de asteptare si in linia 20 care devine

```
20 FOR k = 1 TO 10000
```

R 9.5 Instructiunea BEEP (vezi anexa C) care va fi explicata in capitolul 10.

CAPITOLUL 10
SA INTRODUCEM SUNETE

Dupa cum va amintiti din capitolul 1, CIP-ul este inzestrat cu un difuzor si este capabil sa produca multe melodii pe placul dvs.

Aplicatia practica AP10

1.Introduceti urmatorul program :

```
NEW
 2 PRINT AT 10,1;FLASH1;"ASCULTATI!"
 3 PRINT:PRINT FLASH0;"Cunoasteti melodia?"
 5 FOR n=1 TO 3
10 BEEP .2,11:BEEP .2,12:BEEP .2,14:BEEP .2,11
20 BEEP .2,12:BEEP .2,9:BEEP .2,11:BEEP .2,7
30 BEEP .2,9:BEEP .2,12:BEEP .2,11:BEEP .2,9
40 BEEP .4,14:BEEP .4,14
50 BEEP .2,11:BEEP .2,12:BEEP .2,14:BEEP .2,11
60 BEEP .2,12:BEEP .2,9:BEEP .2,11:BEEP .2,7
70 BEEP .2,9:BEEP .2,12:BEEP .2,11:BEEP .2,9
80 BEEP .4,7:BEEP .4,7
90 NEXT n:CLS
RUN
```

Observati ce se intimpla... Fara a comenta prea mult, ati programat calculatorul sa faca muzica, respectiv sa interpreteze ceva foarte cunoscut nu ?

2.Tastati :

```
NEW
BEEP 1,0
```

Ce se intimpla? Pentru circa 1 secunda, pe ecran nu apare nici o schimbare, in schimb, in microdifuzorul incorporat se aude un sunet in care veti recunoaste nota DO din octava 1 (cei cu ureche muzicala, bineinteles).

Mai incercati :

```
NEW
10 PRINT "BEEP 2,-20":BEEP 2,-20
20 PRINT "BEEP 2,30":BEEP 2,30
30 PRINT "BEEP 4,30":BEEP 4,30:CLS
RUN
```

Sesizati ca inaltimea sunetelor difera, in functie de valoarea celui de al doilea argument din BEEP, iar durata sunetului este data de primul argument.

Notiuni de baza despre producerea sunetelor

Producerea sunetelor se face cu instructiunea BEEP, care are forma generala :

```
BEEP d,i
```

-d (primul argument) reprezinta durata sunetului. Durata in secunde are valori permise intre 0 si 10.

-i (al doilea argument) indica inaltimea sunetului (frecventa), si are

valori permise intre -60 si +69.

Folosirea unor valori in afara limitelor produce o eroare care se soldeaza cu intreruperea executiei si un mesaj de eroare.

Codificarea notelor

Corespondenta intre valorile lui i si gama muzicala (temperata) este data de urmatoarea regula:

Notei DO din octava 1 (DO CENTRAL de pe claviatura unui pian) ii corespunde valoarea 0 mergind in sus (DO#, RE, RE# ...) se creste valoarea lui i cu cite o unitate, mergind in jos (SI, SIb, LA ...) se scade cite o unitate, pina la limitele date mai sus. Precizam ca in gama temperata DO# este aceiasi nota cu REb, etc..., respectiv, diezii si bemolii au ca efect adunarea sau scaderea unei unitati la valoarea corespunzatoare notei (similar ca la pian).

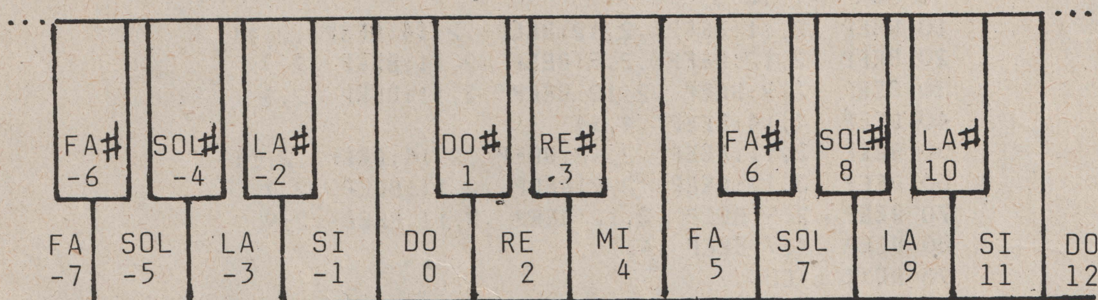


Fig. 10.1

Exemple de programare

Sa incercam pentru inceput sa punem calculatorul sa cinte arpegiul gamei DO MAJOR. Avem nevoie de notele DO, MI, SOL ,DO si sa convenim o durata de .2 secunde pentru toate notele. Programul ar trebui sa fie (sau sa semene cu) :

```
NEW
 5 REM arpegiul gamei DO MAJOR
10 BEEP .2,0
20 BEEP .2,4
30 BEEP .2,7
40 BEEP .2,12
50 BEEP .2,12
60 BEEP .2,7
70 BEEP .2,4
80 BEEP .2,0
RUN
```

Schimbarea octavei

Pentru arpegiul aceleiasi game, dar in alta octava, ar trebui sa adunam sau sa scadem multipli de 12 (sintem siguri ca ati observat ca intre doi DO sint 12 unitati diferenta). Respectiv, pentru octava 2, ar trebui sa folosim valorile 12, 16, 19, 24.

Mai elegant insa ar fi sa scriem programul astfel:
(editati programul anterior, linie cu linie, pentru a-l obtine pe acesta)

```
 5 REM arpegiul gamelor DO MAJOR
10 BEEP .2,0+n
20 BEEP .2,4+n
```

```
30 BEEP .2,7+n
40 BEEP .2,12+n
50 BEEP .2,12+n
60 BEEP .2,7+n
70 BEEP .2,4+n
80 BEEP .2,0
```

si sa-i dam lui n valori de forma ± 12*k avind grija sa nu iesim din limitele permise.

Ca rafinament suplimentar, propunem urmatorul program (procedati la fel, prin editarea programului anterior):

Schimbarea duratei

```
 5 REM arpegiul gamelor DO MAJOR
10 BEEP .2*m,0+n
20 BEEP .2*m,4+n
30 BEEP .2*m,7+n
40 BEEP .2*m,12+n
50 BEEP .2*m,12+n
60 BEEP .2*m,7+n
70 BEEP .2*m,4+n
80 BEEP .2*m,0+n
```

si sa-i dam lui m diverse valori avind deasemeni grija sa nu iesim din gama valorilor permise.

Schimbarea tempou-lui

Am realizat "parametrizarea" programului pentru diverse durate ale notelor si diverse octave. Daca am conveni, de exemplu ca .2 secunde este durata unei optimi, dind diverse valori (intregi sau nu) lui m, am putea interpreta efectul prin schimbarea tempo-ului, sau inlocuirea optimilor cu doimi, note intregi, saisprezecimi, etc...

O utilitate a acestui mod de programare a muzicii ar fi posibilitatea "acordarii" calculatorului cu alte instrumente.

Ca solutie de "codificare" a unei partituri, sugeram sa completati portativul cu o linie dedesubt si una deasupra, si sa scrieti cu creionul in dreptul liniilor si spatiilor (tinind cont de armura cheii) valorile corespunzatoare notelor.

Inca o treaba foarte importanta : cum codificam pauzele ? O solutie a fi folosirea instructiunii PAUSE n, tinind cont ca n produce o pauza de 20*n milisecunde. (Deci PAUSE 50 este echivalent cu BEEP 1...)

Sau altfel, se poate folosi un BEEP cu durata corespunzatoare si o valoare pentru inaltime care sa iasa din spectrul practic audibil (ex. 69). In acest mod uniformizam regula de durata si pentru pauze.
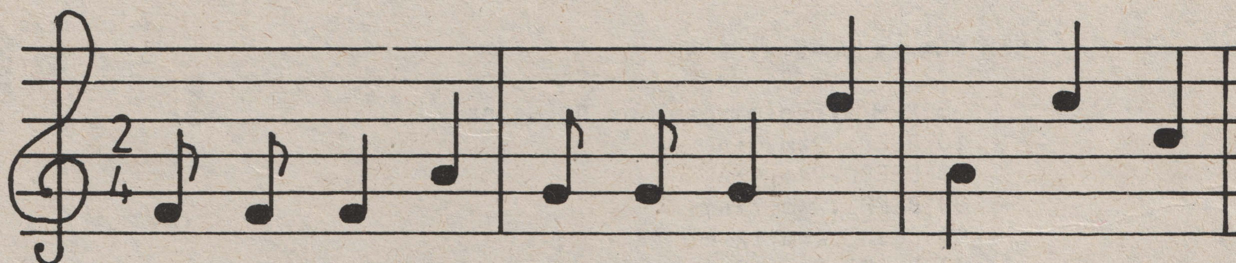
Exemplul 10.2

Sa urmarim programul de mai jos care codifica partitura "Cucule pasare sura" dupa Ioan D. Chirescu.

Iata inceputul:

```
10 BEEP .2,5:BEEP .2,5:BEEP .4,5:BEEP .4,9
20 BEEP .2,7:BEEP .2,7:BEEP .4,7:BEEP .4,12
30 BEEP .4,9:BEEP .4,12:BEEP .4,9
40 BEEP .2,5:BEEP .2,5:BEEP .4,5:BEEP .4,9
50 BEEP .2,7:BEEP .2,7:BEEP .4,7:BEEP .4,12
60 BEEP .4,9:BEEP .4,12:BEEP .4,9
```

```
 70 BEEP .2,12:BEEP .2,12:BEEP .4,12:BEEP .4,14
 30 BEEP .2.10:BEEP .2,10:BEEP .4,10:BEEP .4.12
 85 BEEP .2,9:BEEP .2,9:BEEP .4,9:BEEP .4,5
 90 BEEP .2,7:BEEP .2,7:BEEP .4,7:BEEP .4,12
100 BEEP .4,9:BEEP .4,12:BEEP .4,9
110 BEEP .2,12:BEEP .2,12:BEEP .4,12:BEEP .4,14
120 BEEP .2,10:BEEP .2,10:BEEP .4,10:BEEP .4,12
130 BEEP .2,9:BEEP .2,9:BEEP .4,9:BEEP .4,5
140 BEEP .2,7:BEEP .2,7:BEEP .4,7:BEEP .4,12
145 BEEP .4,5:BEEP.4,12:BEEP .4,9
```



.2,5 .2,5 .4,5 .4,9 .2,7 .2,7 .4,7 .4,12 .4,9 .4,12 .4,9

Fig.10.2

Introduceti programul si cu RUN insarcinati calculatorul cu descifrarea partiturii. Melodia care se aude nu se poate sa nu o recunoasteti. Daca doriti, va puteti folosi la introducere de "parametrizarea" tempo-ului si octavei. Putem folosi ca nume de variabile numele notelor.

```
 5 READ o,p,fa,sol,la,sib,do,re
 8 DATA .2,.4,5,7,9,10,12,14
10 BEEP o,fa:BEEP o,fa:BEEP p,fa:BEEP p,la
20 BEEP o,sol:BEEP o,sol:BEEP p,sol:BEEP p,la
30 BEEP p,la:BEEP p,la:BEEP p,la
40 BEEP o,fa:BEEP o,fa:BEEP p,fa:BEEP p,la
```

Continuati...

Mai facem o precizare pentru cei care doresc sa faca din acest domeniu un cimp de aplicatii mai serioase : exista programe utilitare specializate, asa numitele editoare de muzica cu care se poate ajunge la performante mai ridicate in componistica asistata de calculator.

Deasemeni mai precizam ca la frecvente joase, sunetele ies din clasa celor "muzicale" putind fi folosite pentru efecte sonore interesante.

Intrebari recapitulative si exercitii

I 10.1 Care este forma generala a instructiunii BASIC de prelucrare a sunetelor?

I 10.2 Care sint limitele permise pentru cele doua argumente al instructiunii BEEP?

I 10.3 Carei note ii corespunde valoarea 0 a celui de al doilea argument al instructiunii BEEP?

I 10.4 Care este regula de construire a codificarii notelor, plecind de la corespondenta Do central=0 ?

Probleme

1. Scrieti un program care sa cinte gama DO MAJOR in ambele sensuri.
2. Combinati BEEP cu RND pentru a genera sunete de durate si inaltimi aleatoare, pe un timp nedefinit (intrerupeti cu BREAK!)
3. Codificati urmatoarea partitura:

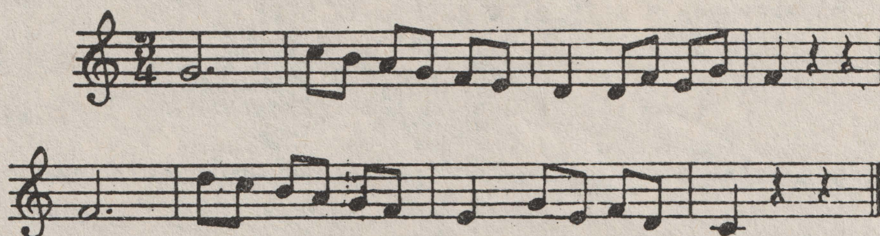


Fig.10.3 (TRIO-W.A.MOZART)

4. Codificati urmatoarea partitura:
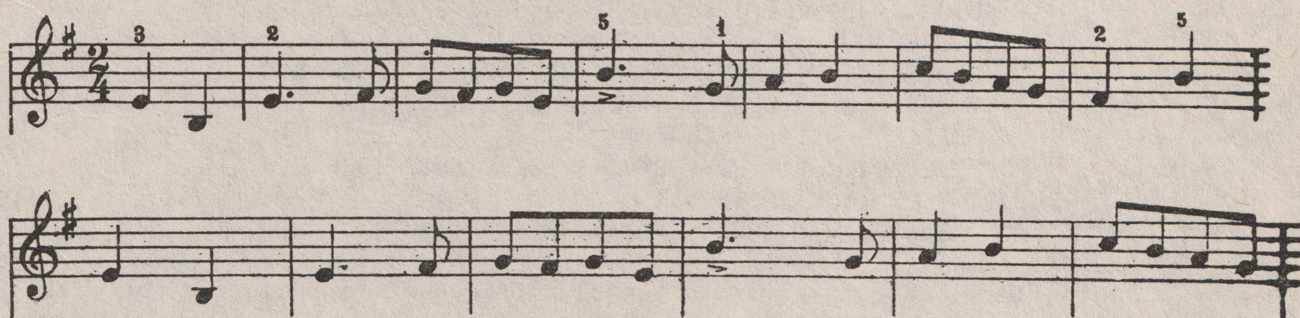


Fig 10.4. (Dupa Arleziana - G. Bizet)

Raspunsuri

R 10.1 BEEP d,i
R 10.2 0 < d < 10 , - 60 < i < +69
R 10.3 DO central
R 10.4 Fata de DO central se aduna 1 spre dreapta si se scade 1 spre stinga, pentru fiecare nota din gama temperata (clapa de pian).

Problema 1

```
10 BEEP  .2,0 :BEEP  .2,2:BEEP .2,4:BEEP.2,5
20 BEEP  .2,7 :BEEP .2,9:BEEP .2,11:BEEP.2,12
30 BEEP  .2,12 :BEEP .2,11:BEEP .2,9:BEEP.2,7
40 BEEP  .2,5 :BEEP  .2,4:BEEP .2,2:BEEP.2,0
```

Problema 2

```
10 BEEP RND*.8, -60 + RND*129
20 GO TO 10
```

Problema 3

```
 5 LET d=...
10 BEEP 3*d,7 : BEEP d,12 : BEEP d,11
20 BEEP  d,9 :BEEP d,7 :BEEP d,5:BEEP d,4
30 BEEP 2*d,2 :BEEP d,2:BEEP d,5 :BEEP d,4
40 BEEP d,7:BEEP 2*d,5:BEEP 2*d,69:BEEP 2*d,69
50 BEEP 3*d,5:BEEP d,14:BEEP d,12:BEEP d,11
60 BEEP  d,9 :  BEEP d,7 :BEEP   d,5
70 BEEP 2*d,4:BEEP d,7:BEEP d,4:BEEP d,5:BEEP d,2
80 BEEP 2*d,0 : BEEP 2*d,69 :BEEP 2*d,69
```

Se va alege pentru d o valoare convenabila pentru durata unei optimi(ex: .3)

Problema 4

```
 5 LET d=...
10 BEEP 2*d,4 : BEEP 2*d,-1
20 BEEP 3*d,4:BEEP d,6:BEEP d,7:BEEP d,6:BEEP d,7
30 BEEP d,4:BEEP 3*d,11:BEEP d,7:BEEP 2*d,9
40 BEEP 2*d,11:BEEP d,12:BEEP d,11:BEEP d,9
45 BEEP d,7:BEEP 2*d,6:BEEP 2*d,11:BEEP d,9
50 BEEP d,7 :BEEP d,6 :BEEP  d,7
60 BEEP 2*d,4 :BEEP 2*d,-1:BEEP 3*d,4 :BEEP d,6
65 BEEP  d,7 :BEEP d,6 :BEEP d,7 :BEEP d,4
70 BEEP 3*d,11 :BEEP d,7
75 BEEP 2*d,9 : BEEP 2*d,11 :BEEP d,12 :BEEP  d,11
80 BEEP  d,9 :BEEP d,7
90 BEEP 2*d,7 :BEEP 2*d,6 :BEEP 4*d,4
```

Se va alege pentru d ca si in exemplul precedent, o valoare convenabila (ex: .3).

CAPITOLUL 11
FUNCTII

BASIC-S va pune la dispozitie anumite functii pe care le puteti folosi in expresii introducind numele functiei urmat de o valoare numit argument.

Efectul este ca in calcularea expresiei respective numele de functie va fi inlocuit cu valoarea functiei corespunzatoare valorii argumentului.

Aplicatia practica AP11

Sa introducem urmatorul program :

```
10 FOR x = 0 TO 2* PI STEP .1
20 PRINT "PENTRU x=";x;" RADIANI"
30 PRINT TAB 4;"SIN(x)=";SIN(x)
40 PRINT TAB4;"COS(x)=";COS(X)
50 NEXT x
RUN
```

Se va observa ca pe ecran apar valorile lui SIN(x) si COS(x).

Deasemeni, se observa ca in linia 10 apare PI, semnificind functia constantei cu acest numar, respectiv: 3,1415927, de asemeni SIN si COS, apelul lor provocind furnizarea unor valori numerice pe care le-am folosit in liniile 10, 30 si 40.

Notiuni de baza

Functiile trigonometrice SIN, COS, TAN, ASN, ACS, ATN:
Forma generala a acestor functii este:

[n] SIN x   sau   [n] SIN(x)

valabila pentru toate cele trei functii:SIN, COS, TAN si inversele lor.

Semnificatia lor este cea cunoscuta din trigonometrie, argumentul functiilor directe este o expresie numerica in radiani. Toate aceste functii se pot utiliza oriunde este acceptata o expresie numerica.

Exemplificare:

```
PRINT SIN(0)        va afisa 0
PRINT SIN(PI/2)     va afisa 1
PRINT SIN PI/2      va afisa 0
PRINT ASN (1)       va afisa 1.5707963
PRINT ASN 1         va afisa 1.5707963
PRINT TAN(ATN(3))   va afisa 3
```

Se observa ca folosirea parantezelor poate schimba valoarea obtinuta. Va recomandam ca atunci cind argumentul este o expresie, aceasta sa fie inclusa intre paranteze, altfel rezultatul poate sa nu fie cel corect.

Functia PI

Forma generala este : PI si are ca efect obtinerea valorii numarului (3.1415927...). Se poate folosi oriunde avem nevoie de o valoare mai exacta a numarului fara a-i scrie noi efectiv atitea zecimale cite avem nevoie. (Calculatorul il genereaza cu 7 zecimale).

Functiile ABS, SGN, INT, SQR, EXP, LN, BIN:

Forma generala pentru ABS:

```
[n] ABS x    sau  [n] ABS (x)
```

valabila pentru toate functiile enumerate, argumentul x fiind un numar sau o expresie numerica.

ABS - returneaza valoarea absoluta a expresiei argument calculata.

Exemple:

```
PRINT ABS (-10)        va afisa 10
PRINT ABS (10)         va afisa 10
PRINT ABS (10-25)      va afisa 15
```

SGN - returneaza :

```
 1 daca argumentul > 0
 0 daca argumentul = 0
-1 daca argumentul < 0
```

Exemple:

```
PRINT SGN (-10)         va afisa -1
LET x=15
PRINT SGN x             va afisa 1
PRINT SGN (-x)          va afisa -1
PRINT SGN (x-x)         va afisa 0
PRINT SGN x-x           va afisa -14   (de ce?)
```

INT - returneaza partea intreaga a expresiei argument

Exemple:

```
INT (3.1415927)          va returna 3
INT (-3.1415927)         va returna -3
```

SQR - extrage radacina patrata .

Exemple:

```
SQR 9               va returna 3
SQR -9              va produce un mesaj de eroare
```

EXP - calculeaza valoarea functiei exponentiale e↑x.

Exemple:

```
PRINT EXP 10        va returna  22026,466,  care  reprezinta
                    puterea a-10-a a numarului e
PRINT EXP 3.14      va returna 23,103867 deci e la puterea 3.14
```

LN - calculeaza valoarea functiei LN x.

Exemple:

```
PRINT LN 10           va afisa 2.3025851
PRINT LN(EXP(10))     va afisa 10
```

BIN xxxxxxxx - aceasta functie specifica faptul ca urmeaza un numar in reprezentarea binara pe un octet. Are ca argument un sir de 8 cifre binare.

POINT

Functia POINT are forma: [n] POINT (x,y) si la apel, returneaza :

0 daca punctul de pe ecran din pozitia x,y are culoarea (INK) identica cu a fundalului (PAPER),

1 in caz contrar.

Functii LEN, STR$, VAL, VAL$, CHR$, CODE

LEN

Functia LEN avind ca argument un sir sau o expresie de tip sir de caractere returneaza numarul caracterelor din sir.

Exemple:

```
PRINT LEN ("ABCDE")            va afisa 5
```

STR$

Functia STR$ are argument numeric si returneaza un sir de caractere care reprezinta numarul.

Exemple:

```
PRINT STR$ 10                  va returna 10
PRINT STR$ (2E3)               va returna 12000
```

VAL

Functia VAL are efectul invers; are ca argument un sir de caractere numerice si returneaza un numar.

Exemple:

```
PRINT VAL "34.756"             va afisa 34.75
PRINT VAL "3E4"                va afisa 30000
PRINT VAL ("2"+"*3"))          va afisa 6
```

VAL$

Functia VAL$ este similara dar are ca rezultat tot un sir.

Exemplu:

```
VAL$ """abcde"""               va afisa "abcde"
```

CHR$

Functia CHR$ are ca argument un numar si returneaza caracterul care are cod acel numar (vezi si anexa A).

Exemplu:

```
PRINT CHR$(42)                 va afisa *
```

CODE

Functia CODE este inversa functiei CHR$. Ea are ca argument un intoarce codul primului caracter al sirului.

Exemplu:

```
PRINT CODE "*"                 va afisa 42
```

INKEY$

Functia INKEY$ returneaza caracterul corespunzator tastei apasate ultima data. Daca nu s-a apasat nici o tasta, se returneaza sirul vid.

Exemple:

```
10 PRINT INKEY$; :PAUSE 10
20 GO TO 10
```

va afisa caracterul de pe tasta pe care ati apasat-o, sau un sir vid.

SCREEN$

Aceasta functie se foloseste in cadrul lui LOAD si SAVE si inlocuieste adresa de inceput a zonei ecran in memorie si numarul de octeti ai acesteia, precedate de CODE. Este utila in LOAD si SAVE.

SCREEN$ <==> CODE 16384,6912

Exemplu:

LOAD "SCREEN$" este echivalent cu LOAD "CODE 16384,6912"

Functii definite de utilizator: DEF FN si FN

In afara de functiile enumerate pe care calculatorul vi le pune la dispozitie, aveti posibilitatea sa va definiti functii proprii,corespunzatoare programelor pe care le scrieti.Pentru aceasta se alege un nume,format dintr-o litera sau dintr-o litera urmata de $ daca functia va returna ca valoare un sir de caractere, si o lista de argumente.

Definirea se face cu:

```
[n] DEF FN nume(x1,x2,...xn)=expresie
```

unde prin expresie se intelege o expresie numerica sau de tip sir de caractere, care poate la rindul ei sa contina apeluri de functii.

Trebuie notat ca argumentele functiei sint in numar de maximum 26 numerice si 26 de tip sir. Deasemenea, ca nota speciala, trebuie retinut ca variabilele functiei sint numite "legate", in sensul ca valabilitatea lor este restrinsa la linia de definire a functiei, neavind nici o legatura cu o alta posibila variabila cu acelasi nume care apare altundeva in program.

De exemplu, in urmatorul program:

```
10 LET x=0:LET y=10
20 LET a=15
30 DEF FN m(x,y)=x+y+x*y+a
40 DEF FN n()=x+y+x*y+a
50 PRINT FN m(2,5)
60 PRINT FN n()
```

La executia liniei 50, valorile 2 si 5 sint trecute pe seama variabilelor x si y din linia 30 fara a altera pe cele din linia 10. Pentru evaluarea functiei m, rezultind valoarea 32 (verificati!), care se afiseaza. La evaluarea liniei 60, se apeleaza functia n definita in linia 40, x si y fiind de data aceasta variabilele libere x si y definite in linia 10 cu valorile 0 si 10 deci rezultatul functiei, desi are aceeasi expresie, va fi altul - cel care se va afisa pe ecran (25).

Exercitii recapitulative

E 11.1 Sa se scrie un program de afisare a valorilor functiilor EXP si LN.

E 11.2 Sa se afiseze pentru fiecare x valorile expresiei:

$$e = \begin{cases} x-1 & \text{pt. } x < 0 \\ 0 & \text{pt. } x = 0 \\ -x+1 & \text{pt. } x > 0 \end{cases}$$

pentru x intre -100 si +100 cu pasul 2.

E 11.3 Folosind functia BIN sa se memoreze intr-o variabila vector codurile binare ale literelor mici de la a la h (vezi anexa la manual), dupa

care sa se afiseze caracterele respective.

E 11.4 Sa se calculeze si sa se afiseze pentru fiecare x folosind functii utilizator valorile functiei:

$$e=\begin{cases} x+0.5\ \ln x & \text{pentru } x\ (0,1] \\ \dfrac{2x+\ln x}{2} & \text{pentru } x\ (1,10] \\ \dfrac{2x\hat{}2+2x+x\ln x+\ln x}{2} & \text{pentru } x>10 \end{cases}$$

E 11.5 Afisati valorile functiei POINT pentru 10 puncte de pe ecran alese aleator.

Probleme

P 11.1 Sa se scrie un program care sa calculeze radacinile ecuatiei de gradul II (toate cazurile).

P 11.2 Sa se calculeze si sa se afiseze pentru fiecare x valorile functiei:

$$f(x)=\begin{cases} \max\{1+x*x, 1-x, \cos x\} & \text{pentru } x<=0 \\ e\uparrow x & \text{pentru } 0<x<=1 \\ 0 & \text{in rest} \end{cases}$$

pentru intervalul -10 si +2 cu pasul .2

P 11.3 Desenati pe ecran 3 cicloide.

Indicatie: Cicloida este definita de ecuatiile

y=r-dcos

x=r+dsin

si se va folosi functia PLOT.

Va recomand: x intre 0 si 22 cu pasul 1/10

r=10, d=20

si in loc de y se va calcula si folosi un y=y+10 pentru a desena cicloida ceva mai sus pe ecran.

Raspunsuri

R 11.1

```
10 FOR x=-100 TO 80 STEP 5
20 LET E=EXP x
30 LET L$="␣"
40 IF x>0 THEN LET L$=" LN(x)="+STR$(LN(x))
50 PRINT "x=";x;" e↑x=";l$
60 NEXT x
```

R 11.2

```
 10 FOR x=-100 TO 100 STEP 2
 20 IF x>0 THEN GO TO 80
 30 IF x=0 THEN GO TO 60
 40 LET e=x-1
 50 GO TO 90
 60 LET x=0
 70 GO TO 90
 80 LET e=-x+1
 90 PRINT "x=";x;" e=";e
100 NEXT x
```

R 11.3

```
 10 DIM a$(8)
 20 LET a$(1)=CHR$ BIN 01100001
 30 LET a$(2)=CHR$ BIN 01100010
 40 LET a$(3)=CHR$ BIN 01100011
 50 LET a$(4)=CHR$ BIN 01100100
 60 LET a$(5)=CHR$ BIN 01100101
 70 LET a$(6)=CHR$ BIN 01100110
 80 LET a$(7)=CHR$ BIN 01100111
 90 LET a$(8)=CHR$ BIN 01101000
100 FOR i=1 to 8
110 PRINT a$(i)
120 NEXT i
```

Liniile 30 la 90 se pot obtine editind linia anterioara modificind numarul de linie, indexul lui a$ si cifrele de la sfirsitul lui BIN.

R 11.4

```
 10 DEF FN f(x)=x+0.5+LNx
 20 DEF FN g(x)=(2*x+(LNx))/2
 30 FOR x=.2 TO 70 STEP .2
 40 IF x>1 THEN GO TO 70
 50 LET e=FN f(x)
 60 GO TO 110
 70 IF x>10 THEN GO TO 100
 80 LET e=FNg(x)
 90 GO TO 110
100 LET e=(x+1)*FNg(x)
110 PRINT "x=␣";x;"␣e=␣";e
120 NEXT x
```

R 11.5

```
 10 FOR I=1 TO 10
 20 LET L=INT(RND*175)
 30 LET C=INT(RND*255)
 40 PRINT "POINT(C,L)="; POINT (C,L)
 50 NEXT I
```

si lansati de mai multe ori cu RUN

P 11.1

```
 10 REM REZOLVAREA ECUATIFI DE GRADUL 2
 20 CLS
 30 PRINT "REZOLVAREA ECUATIEI DE GRADUL 2, DE FORMA ax +bx+c=0"
 40 INPUT "INTRODUCETI VALOAREA a:";a
 45 IF a=0 THEN GO TO 190
 50 INPUT "INTRODUCETI VALOAREA b:";b
 60 INPUT "INTRODUCETI VALOAREA c:";c
 70 LET D=b*b-4*a*c
 80 IF D>=0 THEN GO TO 110
 90 PRINT "ECUATIA NU ARE RADACINI REALE"
100 STOP
110 IF D>0 THEN GO TO 150
120 LET x=-b/(2*a)
130 PRINT "RADACINA UNICA,,x=";x
140 STOP
150 LET x1=(-b+SQRD)/(2*a)
160 LET x2=(-b-SQRD)/(2*a)
```

```
170 PRINT "x1=";x1;"x2=";x2
180 STOP
190 PRINT "a=0 ecuatie degenerata"
200 GO TO 40
```

P 11.2

```
 10 FOR x=-10 TO 2 STEP .2
 20 IF x>0 THEN GO TO 70
 30 LET f=1+x*x
 40 IF f<(1-x) THEN LET f=1-x
 50 IF f<cosx THEN LET f=cosx
 60 GO TO 110
 70 IF x>1 THEN GO TO 100
 80 LET f=EXPx
 90 GO TO 110
100 LET f=0
110 PRINT "x=";x;" f=";f
120 NEXt x
```

P 11.3

```
 10 FOR f=0 TO 23 STEP .2
 20 LET x=10*f-20*SINf
 30 LET y=(10-20*COSf)+10
 40 PLOT INK2, x,y
 50 NEXT f
```

CAPITOLUL 12
CUM LUCRAM DIRECT CU MEMORIA CIP-ULUI

Pentru acei dintre dvs. care au asimilat bine notiunile prezentate pina acum si reusesc sa scrie usor programe in BASIC-S, exista o mare tentatie de a intra tot mai adinc in "tainele" CIP-ului. Pentru aceasta aveti la dispozitie anumite functii / instructiuni care va permit sa lucrati direct cu memoria CIP-ului, sa introduceti informatii direct in ea, sa cititi continutul ei si sa vedeti cum sint reprezentate caracterele etc.

Aplicatia practica Ap12

1. Introduceti:

```
 5 REM program pentru afisarea setului de caractere
10 PRINT "CARACTER","COD":PRINT
20 FOR c=32 TO 255
30 PRINT CHR$ c,c
40 PRINT
50 NEXT c
RUN
```

Notati codurile pentru urmatoarele caractere (priviti Anexa A)

caracter	cod
spatiu (blank)	___
1	___
=	___
A	___
a	___
@	___
RND	___
SIN	___
PEEK	___
STOP	___
RANDOMIZE	___
COPY	___

2. Introduceti si executati:

```
NEW
20 PRINT "x+y="
30 PRINT "adresa","continut"
40 PRINT "------","--------"
50 FOR a=23759 TO 23765
60 PRINT a,PEEKa:PRINT
70 NEXT a
RUN
```

Notati rezultatele si folosind Anexa A scrieti caracterele corespunzatoare codurilor:

adresa	continut	caracter
23759		
23765		

Notiuni de baza

Memoria interna a CIP-ului o putem imagina ca un fiset cu multe sertare numerotate unul dupa altul, incepind cu sertarul avind numarul 0, urmat de sertarul numarul 1, apoi de cel cu numarul 2 s.a.m.d. pina la ultimul sertar de la sfirsitul memoriei.

Un asemenea "sertar" poarta numele de LOCATIE DE MEMORIE si numarul sau reprezinta ADRESA la care se poate memora orice caracter din setul de 255 pe care le cunoaste CIP-ul.

Caracterele se memoreaza prin codurile zecimale cuprinse in Anexa A, dar de fapt in circuite ele sint memorate sub forma binara prin combinatii intre doua cifre:

0 - BIT zero
1 - BIT unu (bit=BInary digiT - cifra binara)

Exemple:

cod zecimal	caracter	binar
32	spatiu	00100000
43	+	00101011
70	F	01000110
102	f	01100110
241	LET	11110001

Observati ca pentru a fi exprimat un caracter se folosesc 8 biti (cifre binare) formind un OCTET (in engleza:BYTE)

Reanalizati cuprinsul memoriei de la AP12 - exercitiul 2:

adresa	cod	caracter	octet
23759	245	PRINT	11110101
23760	34	"	00100010
23761	120	x	01111000
23762	43	+	00101011
23763	121	y	01111001
23764	61	=	00111101
23765	34	"	00100010

Orice octet poate fi localizat in memorie prin ADRESA sau LOCATIA sa.

Memoria CIP-ului poate cuprinde un numar maxim de 65536 octeti primul avind adresa 0, iar ultimul 65535 (pentru ca numaratoarea incepe cu zero).

Capacitatea memoriei se exprima in KILOOCTETI / KILOBYTES 1KO = 1024 octeti

Memoria maxima a CIP-ului este deci de 64 KO.
Memoria calculatorului dvs. este de doua feluri:
RAM (Random Acces Memory) si
ROM (Read Only Memory)
Informatia scrisa in ROM ramine nealterata la decuplarea alimentarii CIP-ului. In schimb, informatia din RAM se pierde la scoaterea de sub tensiune.

Memoria cu BASIC-S

Dupa ce incarcati BASIC-S structura memoriei CIP-ului se schimba. Memoria ROM este invalidata, arhitectura RAM-ului avind urmatorul continut:

```
    |----------|----------------|---------------------|
    |          |                |.Variabile de sistem |
    | BASIC -S | MEMORIE pentru |                     |
    |          | ECRAN          |.programe BASIC      |
    |----------|----------------|---------------------|
adr:0     16383 16384       23295 23296            65535
```

Acces la o locatie de memorie

Pentru inceput sa vedem cum putem avea acces la memorie. Introduceti urmatorul program:

```
10 POKE  32000,128
20 PRINT PEEK 32000
25 STOP
30 POKE 32000,255
40 PRINT PEEK 32000
```

Instructiunea din linia 10 introduce in locatia de memorie cu adresa 32000 numarul 128, iar instructiunea din linia 20 afiseaza continutul locatiei de memorie de la adresa 32000.
Lansarea in executie a programului, cu comanda RUN va avea ca efect afisarea pe ecran a numarului 128.
Executind insa programul cu RUN 30, pe ecran va fi afisata valoarea 255.
Deci s-a modificat informatia scrisa in RAM la adresa 32000 inlocuind valoarea 128 cu 255. Functia POKE scrie in memorie la adresa specificata un octet de informatie, iar functia PEEK extrage de la adresa specificata un octet de informatie.
Sintaxa:

```
POKE adresa,cod
     adresa=0...65535  0<=cod<=255
PEEK adresa
```

RAM VIDEO

Memoria pentru ecran contine informatia care va fi afisata pe ecranul televizorului dvs. Introduceti:

```
10 CLS
20 FOR i=16384 TO 16384 + 6144
30 POKE i, 255
40 PRINT AT 10,10;i
50 NEXT i
RUN
```

Sa lansam in executie programul. Ecranul va incepe sa se umple cu linii

negre si in centrul lui va fi afisata adresa locatiei de memorie ce va fi incarcata cu 255., adica un octet in care toti bitii sint 1. Daca vom modifica linia 30 inlocuind astfel:

```
30 POKE i,15
```

la rularea programului pe ecran va apare o structura de linii albe si negre, octetul 15 avind structura 00001111.

Fiecare caracter este o matrice de 8 * 8 pixeli. Fiecare asemenea matrice are atribute de culoare care sint pozitionate incepind de la dresa 22528 pe o lungime de 22*24. Fiecarui caracter ii este asociat un octet care contine atributele.

Structura octetului de atribute poate fi determinata astfel:

```
atribut = INK + 8 *  PAPER + 64 * BRIGHT + 128 * FLASH
```

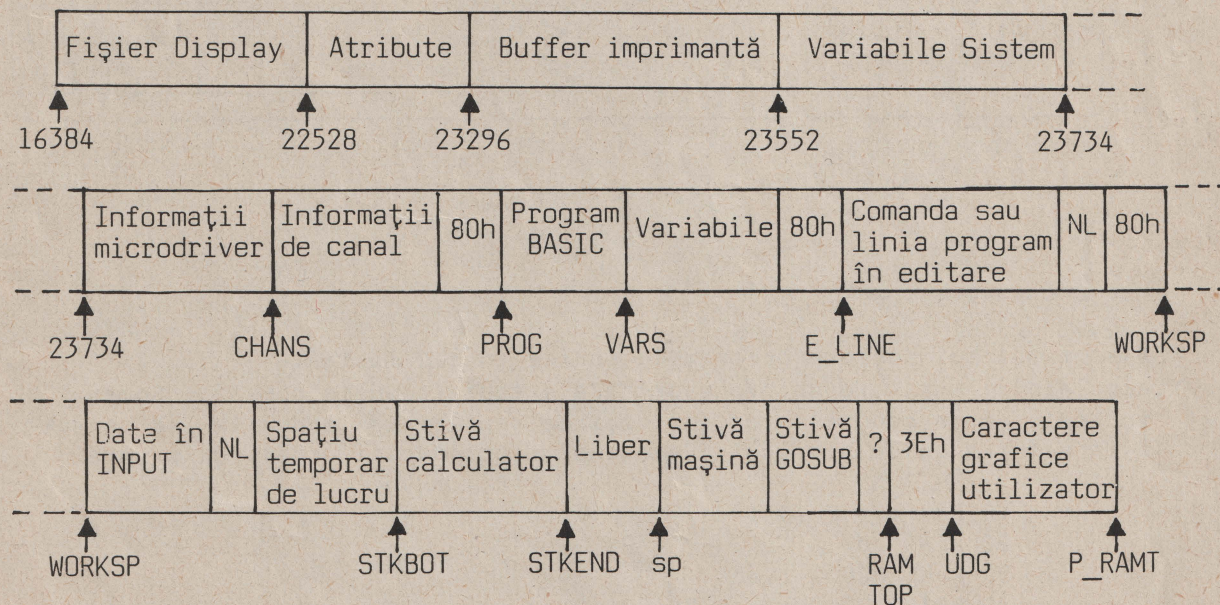
Incercati acum urmatorul program:

```
NEW
10 CLS
20 PRINT AT 10,10;"A"
30 POKE 22528+10*32+10,121
RUN
```

In linia 30 a acestui program am calculat pozitia octetului de atribute pentru caracterul tiparit prin comanda din linia 20.

Variabile de sistem

Pentru a vorbi despre zona variabilelor de sistem sa prezentam mai in amanunt structura memoriei.(Fig. 12.1).



Variabilele de sistem sint explicate in Anexa F.

CLEAR

Instructiunea CLEAR fixeaza noul RAM -TOP, adica adresa maxima alocata

de calculatorul dvs. pentru un program BASIC. Atentie! La o comanda CLEAR se poate sterge continutul memoriei.

In plus, comanda CLEAR are urmatoarele efecte:

-altereaza continuturile variabilelor de sistem;
-sterge VIDEO-RAM
-reseteaza pozitiile PLOT
-executa o comanda RESTORE.

Sintaxa:

```
CLEAR adresa
```

Exemplu:

```
10 CLEAR 40000
RUN
```

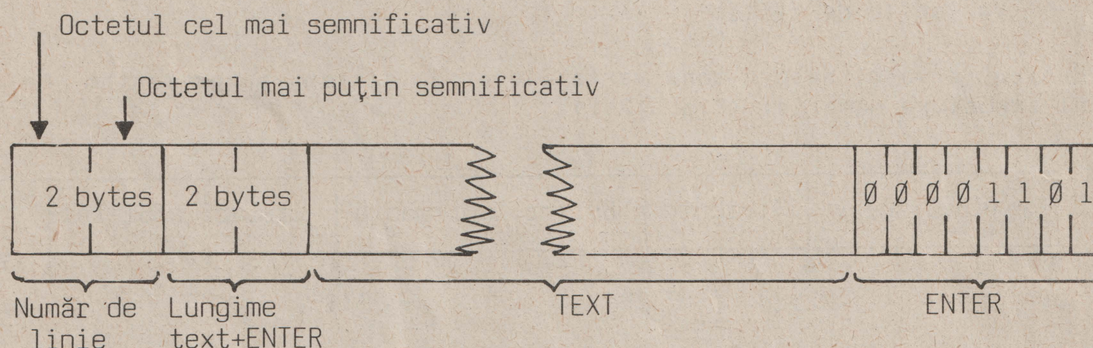
Dupa executia acestei linii zona de memorie alocata pentru programul BASIC este limitata pina la adresa 40000.

NEW

Comanda NEW sterge memoria RAM pina la adresa RAM TOP. Zona UDG ramine neschimbata.

Linie BASIC

Informatia este stocata in memorie dupa diferiti algoritmi. De exemplu in continuare este prezentat modul de organizare a unei linii BASIC in memoria calculatorului.



Cod masina

Si acum mai mult despre programele scrise in cod masina.

Calculatorul dvs. are un procesor care executa anumite comenzi. Comenzile sint indicate prin codurile lor. Octetii ce reprezinta codurile se gasesc stocati in memoria calculatorului. De exemplu introduceti urmatorul program:

```
10 FOR i = 0 TO 17
20 READ adr
30 POKE 32000+i,adr
40 PRINT 32000+i: PEEK(32000+i)
50 NEXT i
60 DATA 62,2,205,1,22,62,22,215,62,10,215,62,10,215,62, 49,
        215, 201
RUN
```

Pe ecran vor apare in stinga adresele de memorie incepind de la 32000 iar in dreapta continutul lor. Cei 18 octeti intre 32000 si 32017 contin o rutina

echivalenta cu:

```
PRINT AT 10,10;1.
```

Pentru a executa programul scris in cod masina se foloseste comanda RANDOMISE USR 32000.

RANDOMISE USR

Sa incercam acum comanda RANDOMISE USR 32000. La executia programului, pe ecran va apare cifra 5.

Acesta a fost un program scris in cod masina. Programatorii experti pot folosi un ASAMBLOR adica un program care preia mnemonicele asociate codurilor de instructiune si le transforma -un format direct executabil.

SAVE CODE

Pentru salvarea pe caseta a rutinelor scrise in cod masina se foloseste instructiunea:

SAVE "nume" CODE adresa,nr.octeti

LOAD CODE

Pentru incarcarea programului in memorie: LOAD "nume program" CODE adresa de incarcare

Daca doriti verificarea unui program salvat introduceti:

VERIFY "nume" CODE

Lucrul cu periferce

Si acum sa vedem cum comunica un calculator cu lumea inconjuratoare.

Ce este un PORT intrare/iesire

Microprocesorul Z80 considera perifericele fie ca o zona de memorie (RAM-VIDEO - pentru ecran), fie ca un port de intrare (tastatura, casetofon) sau de iesire (BORDER, DIFUZOR, CASETOFON).

Daca la RAM-VIDEO am vazut cum puteti avea acces, sa discutam modul de acces al CIP-ului la porturile de intrare/iesire.

Calculatorul dvs. poate avea in configuratie 256 porturi notate de la 0 la 255.

In arhitectura hardware a CIP-ului un alt rol important are portul 238 care permite validarea / invalidarea ROM-ului.

OUT si In

Introduceti:

```
NEW
10 INPUT  "culoarea BORDER?" ,i
20 OUT 254,i
30 GO TO 10
RUN
```

Alegind pentru variabila i valori intre 0 si 7 vom putea modifica culoarea BORDER-ului.

Observati schimbarea BORDER-ului!

```
NEW
10 INPUT i
```

```
20 OUT 254,10
30 OUT 254,0
40 PAUSE i
50 GO TO 10
RUN
```

Daca se atribuie lui i valoarea 50 si difuzorul va emite un biriit la fiecare secunda. Daca i se modifica, se va schimba corespunzator si perioada semnalului sonor.

Instructiunea pentru transmiterea unei date la un port de iesire are

SINTAXA: OUT adresa,octet   adresa=adresa portului

Pentru a citi de la un port de intrare-iesire se foloseste instructiunea:

```
IN adresa
```

Intrebari recapitulative

I 12.1 Care este variabila de sistem care contine adresa ultimului octet din zona RAM adresabila de procesor. (Consultati Anexa F).

I 12.2 Scrieti un program in cod masina care sa incarce locatia de memorie 17000 cu valoarea 255. Ce observati la executarea programului? Cum puteti modifica atributele zonei de RAM-VIDEO care a fost modificata la executia programului?

I 12.3 Salvati un ecran :
a) fara zone de atribute video
b) cu zone de atribute

I 12.4 Scrieti un program care sa creeze in zona BORDER o succesiune de linii verzi si rosii.

Raspunsuri

R 12.1 Variabila este P - RAMT si este localizata pe 2 octeti la adresele 23732 si respectiv 23733.

R 12.2 10 POKE 17000,255
Pentru a afla octetul care contine informatia de atribute locatiei la adresa 17000 din VIDEO-RAM folosim urmatorul program:

```
10 REM modificare informatie din VIDEO-RAM
20 POKE 17000,255
30 REM PROGRAM CARE URMARESTE MODIFICAREA ZONEI DE ATRIBUTE
40 FOR i=16384+6144 TO 16384+6912
50 POKE i,
60 PRINT AT 10,10;i
70 PAUSE 0
80 REM  CONTINUA DOAR DACA APASAM ORICE TASTA
90 NEXT i
```

Locatia cautata are adresa 22632.

R 12.3 a) Pentru zona de VIDEO-RAM fara atribute se foloseste:

```
10 SAVE "1" CODE 16384,6144
```

b)Pentru zona de VIDEO-RAM cu atribute se foloseste:

```
10 SAVE "2" CODE 16384,6912
```

R 12.4

```
10 REM BORDER VERDE
20 OUT 234,4
30 REM BORDER ROSU
40 OUT 234,2
50 GO TO 20
```

ANEXA A

Code	Caracter	Hexa	Mnemonic	-dupa CB	- dupa ED
0		00	nop	rlc b	
1		01	ld bc,NN	rlc c	
2		02	ld(bc),a	rlc d	
3	nefolosite	03	inc bc	rlc e	
4		04	inc b	rlc h	
5		05	inc b	rlc l	
6	PRINT virgula	06	ld b,N	rlc(hl)	
7	EDIT	07	rlca	rlc a	
8	cursor stinga	08	ex af,af´	rrc b	
9	cursor dreapta	09	add hl,bc	rrc c	
10	cursor jos	0A	ld a,(bc)	rrc d	
11	cursor sus	0B	dec bc	rrc e	
12	DELETE	0C	inc c	rrc h	
13	ENTER	0D	dec c	rrc l	
14	numar	0E	ld c,N	rrc(hl)	
15	nefolosit	0F	rrca	rrc a	
16	INK control	10	djnz e	rl b	
17	PAPER control	11	ld de, NN	rl c	
18	FLASH control	12	ld(de),a	rl d	
19	BRIGHT control	13	inc de	rl e	
20	INVERSE control	14	inc d	rl h	
21	OVER control	15	dec d	rl l	
22	AT control	16	ld d,N	rl(hl)	
23	TAB control	17	rla	rl a	
24		18	jr e	rr b	
25		19	add hl,de	rr c	
26		1A	ld a,(de)	rr d	
27	nefolosite	1B	dec de	rr e	
28		1C	inc e	rr h	
29		1D	dec e	rr l	
30		1E	ld e,N	rr(hl)	
31		1F	rra	rr a	
32	spatiu	20	jr nz,e	sla b	
33	!	21	ld hl,NN	sla c	
34	"	22	ld(NN),hl	sla d	
35	#	23	inc hl	sla e	
36	$	24	inc h	sla h	
37	%	25	dec h	sla l	
38	&	26	ld h,N	sla(hl)	
39	'	27	daa	sla a	
40	(	28	jr z,e	sra b	
41	)	29	add hl,hl	sra c	
42	*	2A	ld hl,(NN)	sra d	
43	+	2B	dec hl	sra e	
44	,	2C	inc l	sra h	
45	-	2D	dec l	sra l	
46	.	2E	ld l,N	sra(hl)	
47	/	2F	cpl	sra a	
48	0	30	jr nc,e		
49	1	31	ld sp,NN		
50	2	32	ld (NN),a		
51	3	33	inc sp		
52	4	34	inc (hl)		
53	5	35	dec(hl)		
54	6	36	ld(hl),N		

55	7	37	scf		
56	8	38	jr c,e	srl b	
57	9	39	add hl,sp	srl c	
58	:	3A	ld a,(NN)	srl d	
59	;	3B	dec sp	srl e	
60	<	3C	inc a	srl h	
61	=	3D	dec a	srl l	
62	>	3E	ld a,N	srl (hl)	
63	?	3F	ccf	srl a	
64	@	40	ld b,b	bit 0,b	in b,(c)
65	A	41	ld b,c	bit 0,c	out(c),b
66	B	42	ld b,d	bit 0,d	sbc hl,bc
67	C	43	ld b,e	bit 0,e	ld(NN),bc
68	D	44	ld b,h	bit 0,h	neg
69	E	45	ld b,l	bit 0,l	retn
70	F	46	ld b,(hl)	bit 0(hl)	im 0
71	G	47	ld b,a	bit 0,a	ld i,a
72	H	48	ld c,b	bit 1,b	in c(c)
73	I	49	ld c,c	bit 1,c	out (c),c
74	J	4A	ld c,d	bit 1,d	adc hl,bc
75	K	4B	ld c,e	bit 1,e	ld bc(NN)
76	L	4C	ld c,h	bit 1,h	
77	M	4D	ld c,l	bit 1,l	reti
78	N	4E	ld c,(hl)	bit 1,(hl)	
79	O	4F	ld c,a	bit 1,a	ld r,a
80	P	50	ld d,b	bit 2,b	in d,(c)
81	Q	51	ld d,c	bit 2,c	out(c),d
82	R	52	ld d,d	bit 2,d	sbc hl,de
83	S	53	ld d,e	bit 2,e	ld(NN),de
84	T	54	ld d,h	bit 2,h	
85	U	55	ld d,l	bit 2,l	
86	V	56	ld d,(hl)	bit 2,(hl)	im 1
87	W	57	ld d,a	bit 2,a	ld a,i
88	X	58	ld e,b	bit 3,b	in e,(c)
89	Y	59	ld e,c	bit 3,c	out(c),e
90	Z	5A	ld e,d	bit 3,d	adc hl,de
91	[	5B	ld e,e	bit 3,e	ld de(NN)
92	/	5C	ld e,h	bit 3,h	
93	]	5D	ld e,l	bit 3,l	
94	↑	5E	ld e,(hl)	bit 3,(hl)	im 2
95	_	5F	ld e,a	bit 3,a	ld a,r
96	£	60	ld h,b	bit 4,b	in h,(c)
97	a	61	ld h,c	bit 4,c	out(c),h
98	b	62	ld h,d	bit 4,d	sbc hl,hl
99	c	63	ld h,e	bit 4,e	ld(NN),hl
100	d	64	ld h,h	bit 4,h	
101	e	65	ld h,l	bit 4,l	
102	f	66	ld h,(hl)	bit 4,(hl)	
103	g	67	ld h,a	bit 4,a	rrd
104	h	68	ld l,b	bit 5,b	in l,(c)
105	i	69	ld l,c	bit 5,c	out(c),l
106	j	6A	ld l,d	bit 5,d	adc hl,hl
107	k	6B	ld l,e	bit 5,e	ld hl(NN)
108	l	6C	ld l,h	bit 5,h	
109	m	6D	ld l,l	bit 5,l	
110	n	6E	ld l,(hl)	bit 5,(hl)	
111	o	6F	ld l,a	bit 5,a	rld
112	p	70	ld(hl),b	bit 6,b	in f,(c)
113	q	71	ld(hl),c	bit 6,c	

114	r		72	ld(hl),d	bit 6,d	sbc hl,sp
115	s		73	ld(hl),e	bit 6,e	ld(NN),sp
116	t		74	ld(hl),h	bit 6,h	
117	u		75	ld(hl),l	bit 6,l	
118	v		76	halt	bit 6,(hl)	
119	w		77	ld(hl),a	bit 6,a	
120	x		78	ld a,b	bit 7,b	in a,(c)
121	y		79	ld a,c	bit 7,c	out (c),a
122	z		7A	ld a,d	bit 7,e	ld sp,(NN)
124	¦		7C	ld a,h	bit 7,h	
125	}		7D	ld a,l	bit 7,l	
126	~		7E	ld a,(hl)	bit 7,(hl)	
127	©		7F	ld a,a	bit 7,a	
128	▢		80	add a,b	res 0,b	
129	▝		81	add a,c	res 0,c	
130	▘		82	add a,d	res 0,d	
131	▀		83	add a,e	res 0,e	
132	▗		84	add a,h	res 0,h	
133	▐		85	add a,l	res 0,l	
134	▚		86	add a,(hl)	res 0,(hl)	
135	▜		87	add a,a	res 0,a	
136	▖		88	adc a,b	res 1,b	
137	▞		89	adc a,c	res 1,c	
138	▌		8A	adc a,d	res 1,d	
139	▛		8B	adc a,e	res 1,e	
140	▄		8C	adc a,h	res 1,h	
141	▟		8D	adc a,l	res 1,l	
142	▙		8E	adc a,(hl)	res 1,(hl)	
143	█		8F	adc a,a	res 1,a	
144	(a)		90	sub b	res 2,b	
145	(b)		91	sub c	res 2,c	
146	(c)		92	sub d	res 2,d	
147	(d)		93	sub e	res 2,e	
148	(e)		94	sub h	res 2,h	
149	(f)		95	sub l	res 2,l	
150	(g)		96	sub (hl)	res 2,(hl)	
151	(h)		97	sub a	res 2,a	
152	(i)		98	sbc a,b	res 3,b	
153	(j)	caractere	99	sbc a,c	res 3,c	
154	(k)	grafice	9A	sbc a,d	res 3,d	
155	(l)	utilizator	9B	sbc a,e	res 3,e	
156	(m)		9C	sbc a,h	res 3,h	
157	(n)		9D	sbc a,l	res 3,l	
158	(o)		9E	sbc a,(hl)	res 3,(hl)	
159	(p)		9F	sbc a,a	res 3,a	
160	(q)		A0	and b	res 4,b	ldi
161	(r)		A1	and c	res 4,c	cpi
162	(s)		A2	and d	res 4,d	ini
163	(t)		A3	and e	res 4,e	outi
164	(u)		A4	and h	res 4,h	
165	RND		A5	and l	res 4,l	
166	INKEY$		A6	and (hl)	res 4,(hl)	
167	PI		A7	and a	res 4,a	
168	FN		A8	xor b	res 5,b	ldd
169	POINT		A9	xor c	res 5,c	cpd
170	SCREEN$		AA	xor d	res 5,d	ind
171	ATTR		AB	xor e	res 5,e	outd
172	AT		AC	xor h	res 5,h	
173	TAB		AD	xor l	res 5,l	

174	VAL$	AE	xor (hl)	res 5,(hl)	
175	CODE	AF	xor a	res 5,a	
176	VAL	B0	or b	res 6,b	ldir
177	LEN	B1	or c	res 6,c	cpir
178	SIN	B2	or d	res 6,d	inir
179	COS	B3	or e	res 6,e	otir
180	TAN	B4	or h	res 6,h	
181	ASN	B5	or l	res 6,l	
182	ACS	B6	or (hl)	res 6,(hl)	
183	ATN	B7	or a	res 6,a	
184	LN	B8	cp b	res 7,b	lddr
185	EXP	B9	cp c	res 7,c	cpdr
186	INT	BA	cp d	res 7,d	indr
187	SQR	BB	cp e	res 7,e	otdr
188	SGN	BC	cp h	res 7,h	
189	ABS	BD	cp l	res 7,l	
190	PEEK	BE	cp (hl)	res 7,(hl)	
191	IN	BF	cp a	res 7,a	
192	USR	C0	ret nz	set 0,b	
193	STR$	C1	pop bc	set 0,c	
194	CHR$	C2	jp nz,NN	set 0,d	
195	NOT	C3	jp NN	set 0,e	
196	BIN	C4	call nz,NN	set 0,h	
197	OR	C5	push bc	set 0,l	
198	AND	C6	add a,N	set 0,(hl)	
199	<=	C7	rst 0	set 0,a	
200	>=	C8	ret z	set 1,b	
201	<>	C9	ret	set 1,c	
202	LINE	CA	jp z,NN	set 1,d	
203	THEN	CB		set 1,e	
204	TO	CC	call z,NN	set 1,h	
205	STEP	CD	call NN	set 1,l	
206	DEF FN	CE	adc a,N	set 1,(hl)	
207	CAT	CF	rst 8	set 1,a	
208	FORMAT	D0	ret nc	set 2,b	
209	MOVE	D1	pop de	set 2,c	
210	ERASE	D2	jp nc,NN	set 2,d	
211	OPEN#	D3	out (N),a	set 2,e	
212	CLOSE#	D4	call nc,NN	set 2,h	
213	MERGE	D5	push de	set 2,l	
214	VERIFY	D6	sub N	set 2,(hl)	
215	BEEP	D7	rst 16	set 2,a	
216	CIRCLE	D8	ret c	set 3,b	
217	INK	D9	exx	set 3,c	
218	PAPER	DA	jp c,NN	set 3,d	
219	FLASH	DB	IN a,(N)	set 3,e	
220	BRIGHT	DC	call c,NN	set 3,h	
221	INVERSE	DD	pref.instr. using ix	set 3,l	
222	OVER	DE	sbc a,N	set 3,(hl)	
223	OUT	DF	rst 24	set 3,a	
224	LPRINT	E0	ret po	set 4,b	
225	LLIST	E1	pop hl	set 4,c	
226	STOP	E2	jp po,NN	set 4,d	
227	READ	E3	ex (sp),hl	set 4,e	
228	DATA	E4	call po,NN	set 4,h	
229	RESTORE	E5	push hl	set 4,l	
230	NEW	E6	and N	set 4,(hl)	
231	BORDER	E7	rst 32	set 4,a	

232	CONTINUE	E8	ret pe	set 5,b
233	DIM	E9	jp (hl)	set 5,c
234	REM	EA	jp pe,NN	set 5,d
235	FOR	EB	ex de,hl	set 5,e
236	GOTO	EC	call pe,NN	set 5,h
237	GOSUB	ED		set 5,l
238	INPUT	EE	xor N	set 5,(hl)
239	LOAD	EF	rst 40	set 5,a
240	LIST	F0	ret p	set 6,b
241	LET	F1	pop af	set 6,c
242	PAUSE	F2	jp p,NN	set 6,d
243	NEXT	F3	di	set 6,e
244	POKE	F4	call p,NN	set 6,h
245	PRINT	F5	push af	set 6,l
246	PLOT	F6	or N	set 6,(hl)
247	RUN	F7	rst 48	set 6,a
248	SAVE	F8	ret m	set 7,b
249	RANDOMIZE	F9	ld sp,hl	set 7,c
250	IF	FA	jp m,NN	set 7,d
251	CLS	FB	ei	set 7,e
252	DRAW	FC	call m,NN	set 7,h
253	CLEAR	FD	pref. instr. using iy	set 7,l
254	RETURN	FE	cp N	set 7,(hl)
255	COPY	FF	rst 56	set 7,a

CARACTER	TASTATURA	CARACTER	TASTATURA
ABS	(CS+SS) G	CLOSE#	(CS+SS) CS+5
ACS	(CS+SS) SS+W	:CLS	V
AND	SS+Y	CODE	(CS+SS) I
ASN	(CS+SS) SS+Q	:CONTINUE	C
AT	SS+I	:COPY	Z
ATN	(CS+SS) SS+E	CYAN	5
ATTR	(CS+SS) SS+L		
		DATA	(CS+SS) D
BEEP	(CS+SS) SS+Z	DEF FN	(CS+SS) SS+1
BIN	(CS+SS) B	DELETE	CS+0
BLACK	0	DIM	D
BLUE	1	DRAW	W
:BORDER	B		
BREAK	(CS+SPACE)	EDIT	(CS+1)
BRIGHT	(CS+SS) SS+B	ENTER	CR
		ERASE	(CS+SS) SS+7
CAT	(CS+SS) SS+9	EXP	(CS+SS) X
CHR$	(CS+SS) U		
CIRCLE	(CS+SS) SS+H	FLASH	(CS+SS) SS+V
:CLEAR	X	FN	(CS+SS) SS+2
:FOR	F	LLIST	(CS+SS) V
FORMAT	(CS+SS) SS+0	LN	(CS+SS) Z
		:LOAD	J
:GO SUB	H	LPRINT	(CS+SS) C
:GO TO	G		
GRAPHICS	(CS+9)	MERGE	(CS+SS) SS+T
GREEN	4	MOVE	(CS+SS) SS+6
:IF	U	:NEW	A
IN	(CS+SS) SS+I	:NEXT	N
INK	(CS+SS) SS+X	NOT	SS+S
INKEY$	(CS+SS) N		
:INPUT	I	:OPEN#	(CS+SS) SS+4
INT	(CS+SS) R	OR	SS+U
INVERSE	(CS+SS) SS+M	OUT	(CS+SS) SS+0
INVIDEO	(CS+4)	OVER	(CS+SS) SS+N
:LEN	(CS+SS) K	:PAPER	(CS+SS) SS+C
:LET	L	PAUSE	M
LINE	(CS+SS) SS+3	PEEK	(CS+SS) O
:LIST	K	PI	(CS+SS) M
:PLOT	Q	TAB	(CS+SS) P
POINT	(CS+SS) SS+8	TAN	(CS+SS) E
:POKE	O	THEN	SS+G
:PRINT	P	TO	SS+F
:RAND	T	TR.VIDEO	(CS+3)
READ	(CS+SS) A		
RED	2	USR	(CS+SS) L
:REM	E		
RESTORE	(CS+SS) S	VAL	(CS+SS) J
RETURN	Y	VAL$	(CS+SS) SS+J
RND	(CS+SS) T	VERIFY	(CS+SS) SS+R

:RUN		R
:SAVE		S
SCREEN$	(CS+SS)	SS+K
SGN	(CS+SS)	F
SIN	(CS+SS)	Q
SQR	(CS+SS)	H
STEP		SS+D
STOP		SS+A
STR$	(CS+SS)	Y
▚	(CS+9)	6
▜	(CS+9)	7
▖	(CS+9)	CS+7
▞	(CS+9)	CS+6
▌	(CS+9)	CS+5
▛	(CS+9)	CS+4
▄	(CS+9)	CS+3
▟	(CS+9)	CS+2
▙	(CS+9)	CS+1
█	(CS+9)	CS+8
!		SS+1
" (ghilimele)		SS+P
#		SS+3
$		SS+4
%		SS+5
&		SS+6
′ (apostrof)		SS+7
(		SS+8
)		SS+9
* (asterisc)		SS+B
~	(CS+SS)	SS+A
©	(CS+SS)	SS+P
<=		SS+Q
>=		SS+E
<>		SS+W
cursor sus	CS+7	
cursor dr.	CS+8	
cursor st.	CS+5	
cursor jos	CS+6	

video invers	(cs+4)	
WHITE		
		7
□	(CS+9)	8
▝	(CS+9)	1
▘	(CS+9)	2
▀	(CS+9)	3
▗	(CS+9)	4
▐	(CS+9)	5
+		SS+K
, (virgula)		SS+N
- (minus)		SS+J
. (punct)		SS+M
/ (slash)		SS+B
:		SS+Z
;		SS+O
<		SS+R
=		SS+L
>		SS+T
?		SS+C
@ (a rond)		SS+2
[	(CS+SS)	SS+Y
\ (slash inv.)	(CS+SS)	SS+D
]	(CS+SS)	SS+U
↑		SS+H
_ (subl.)		SS+0
(lira)		SS+X
{ (acolada)	(CS+SS)	SS+F
¦ (bara vert.)	(CS+SS)	SS+S
} (acolada)	(CS+SS)	SS+G

LIMBAJUL BASIC-S
(MEMENTO)

-Gama numerelor intregi:
- 32000.... +32000

-Gama numerelor reale:
4*10↑-39....10↑38

-Constante numerice

.format intreg
ex.: 375

.format real
```
ex.:375.6
       ↑
       └── virgula zecimala
```

.format exponential:
```
ex.: 3.6245E3  (=3624,5)
     3.62E-2   (=0,0036)
     ---- --
mantisa ──↑    ↑──exponent
```

-Constante sir carcatere:
ex.:"ABC"   "A1-C"

-Variabile simple - o litera urmata sau nu de una sau mai multe litere/cifre

-Variabile indexate - tablouri (matrici) - litera urmata de dimensiuni in paranteze

ex.: V(3)   V(3,7)

-Variabile sir - nume variabila urmat de $

ex.: A$ z1$ A$7 z1$(I,K)

-Operatori

.aritmetici: + - * / ↑
.relationali: =,<,>,<=,>=,<>
.logici: AND OR NOT
.de concatenare siruri: +

-Prioritatea in executarea operatiilor (descrescatoare):

↑ (ridicare la putere)
*,/ (inmultire, impartire)
+.- (adunare, scadere)
=,>,<,<=,>=,<>
NOT
AND
OR

Comenzi, instructiuni, functii (dupa cerinte de utilizare)

-Comenzi de executie program:

.CLEAR .LIST
.CLS (CLEAR SCREEN) .NEW
.CONT (CONTINUE) .RUN

-Comenzi pentru lucru cu caseta magnetica:

.LOAD .SAVE
.MERGE .VERIFY

-Comenzi pentru lucru cu imprimanta:

.COPY (OPEN# si CLOSE#)
.LLIST
.LPRINT

-Instructiuni de intrare-iesire

.INPUT .PRINT AT

```
        .LET          .READ, DATA, RESTORE
        .PRINT
    -Instructiuni de control
        .GO TO              .FOR, NEXT
        .IF...THEN          .GOSUB, RETURN
    -Instructiune pentru lucru cu tablouri (matrici)
        .DIM
    -Instructiune pentru comentarii
        .REM
    -Instructiuni grafice:
        .CIRCLE          .PLOT
        .DRAW
    -Instructiuni de utilizare a culorilor:
        .ATTR        .INK
        .BORDER      .INVERSE
        .BRIGHT      .OVER
        .FLASH       .PAPER
    -Instructiune pentru sunete:
        .BEEP
    -Instructiune de definire functie utilizator:
        .DEF FN
    -Instructiuni de oprire executie program:
        .PAUSE
        .STOP
    -Functii standard:
        .ABS         .PEEK
        .ACS         .PI
        .AND         .POINT
        .ASM         .RND
        .ATM         .SCREEN$
        .BIN         .SGN
        .CHR$        .SIN
        .CODE        .SQR
        .COS         .STR$
        .EXP         .TAN
        .FN          .USR
        .IN          .VAL
        .INKEY#      .VAL$
        .INT
        .LEN
        .LN
        .NOT
        .OR
```

Comenzi, functii instructiuni (in ordine alfabetica)

```
ABS numar - calculeaza valoarea absoluta a numarului
ACS numar - calculeaza arccosinusul numarului (radiani)
            xANDy - daca: y=0 :xANDy = x
                          y=0  xANDy = 0
            x$ANDy - daca y=o  x$ANDy = x$
                          y=0 x$ANDy ="" (sir nul)
ASN numar - calculeaza arcsinusul numarului (radiani)
ATN numar - calculeaza arctangenta numarului (radiani)
ATTR (linie,coloana) - da informatii despre atributele  de  culoare  ale
                       unui caracter
BEEP x,y - scoate o nota muzicala de inaltime y cu o durata de x secunde
BIN biti - indica un numar binar
BORDER n - coloreaza marginea imaginii cu culoarea data de valoarea lui n
```

(n=0 la 7)

BRIGHT n - produce afisarea caracterelor: normal (n=0), stralucitor (n=1)

CHR$ - da caracterul al carui cod este x, sub forma de sir ("caracter")

CIRCLE x,y,r - deseneaza un cerc de raza "r" cu centrul in punctul de coordonate "x,y"

CLEAR n - sterge din memorie toate variabilele introduse anterior prin program si modifica variabila de sistem RAMTOP la adresa n

CLS - sterge ecranul si anuleaza continutul memoriei ecran, fara a influenta variabilele stabilite prin program

CODE x$ - da codul primului caracter din sirul x

CONT - continua un program oprit temporar prin STOP (CONTINUE)

COPY - copiaza primele 22 de linii ale ecranului pe imprimanta

COS x - calculeaza cosinusul unghiului x (x in radiani)

DATA c1, c2,... - introduce valori (constante numerice / sir de caractere) pentru variabilele folosite in program.

Ex.:135 DATA 3.2, "ABC",43

DEF FN n(V1,V2...)=e - defineste o functie nestandard, a utilizatorului
DEF FN n$(V1$, V2$...)=e

.n este un numar dat printr-o litera
.V1, V2... sint nume de variabile constituind parametrii functiei (maxim 26)
.e este o expresie matematica

```
Ex.: 10 DEF FN p(x,y)=x↑2*3
     70 LET A=FN p(3,7)
```

DIM a(n1,n2...) - declara tablouri sau (matrici) de numere sau de siruri
DIM a$(n1,n2...)

.a/a$ este numele tabloului
.n1,n2... sint dimensiunile tabloului

```
Ex.:DIM b(14)     DIM m(3,15)
    DIM c$(7)     DIM g$(7,15)
```

DRAW x,y,z - deseneaza o linie de la pozitia curenta a cursorului grafic pina la punctul de coordonate x si y pixeli

-255<=x<=255   -175<=y<=175
.z este unghiul (in radiani) pentru descrierea unei linii circulare

```
Ex.:DRAW 255,0  DRAW x-45,y-80
    DRAW 60,60,PI
```

EXP x - calculeaza functia exponentiala e la puterea x

FLASH n - produce afisarea normala (n=0) sau clipitoare (n=1)

FOR x = n TO m - executa secventa de instructiuni care o urmeaza
FOR x = n TO m STEP y (pina la NEXTx) de un numar de ori dat de relatia:

(m-n)/(y-1)

.x este numele variabilei (o litera)
.n, m si y sint numere sau expresii aritmetice

```
Ex. : 30 FOR i=1 TO 12 STEP 5
      40 FOR j=14 TO 3 STEP -2
      :
      :
      80 NEXT j:NEXT i
```

GO SUB nr. linie - produce saltul in program la subrutina care incepe cu linia indicata si se termina cu prima instructiune RETURN

Dupa executarea subrutinei programul se reia cu instructiunile dupa GOSUB

Ex.:60 GO SUB 9000

GO TO n - produce salt neconditionat la linia n

Ex.:IF x=0 THEN GOTO 10

IF x THEN s - daca expresia x este adevarata (diferita de zero) se executa s, in caz contrar se executa instructiunea care urmeaza dupa IF.

Ex.:IFa$="SF" AND b=0 PRINT b
IF 1=0 THEN GOTO 100

IN m - citeste din memorie octetul din portul cu adresa m

INK n - determina culoarea cu care vor fi afisate caracterele ce urmeaza (n=0 la 7)

INKEY$ - introduce caracterul primit de la tastatura
Ex.:LET a$=INKEY$: IF INKEY$="N" THEN GO TO 80

INPUT "c",v1,v2...- permite introducerea datelor de la o tastatura sau
INPUT LINE...
.c este un sir de caractere optionale care expliciteaza datele ce vor fi introduse
.v1,v2... sint nume de variabile numerice sau sir de caractere

```
Ex.:INPUT R
    INPUT "INTRODUCETI PRETUL";p
    INPUT n$
    INPUT LINE a$
```

INT n - da o valoare intreaga a numarului n (<=n)

INVERSE n - controleaza inversarea afisarii caracterelor ce urmeaza
.daca n=0 caracterele sint afisate video normal
.daca n=1 caracterele sint afisate video invers

LEN sir - da numarul de caractere continute in sir

LET v=e - atribuie variabilei v valoarea expresiei e

```
Ex.:LET i=i+1
    LET a=32
    LET N$="NUME"
    LET SUMA=4.25*A 2
```

LIST - listeaza programul, incepind cu prima linie

LIST n - listeaza programul incepind cu prima linie al carui numar >=n; linia n devine linie curenta

LLIST/LLIST n - listeaza programul pe imprimanta

LN numar - calculeaza logaritmul natural al numarului

LOAD "nume" - incarca in memorie programul cu numele specificat
LOAD "nume" CODE adresa, numar
- incarca un numar de octeti incepind de la adresa specificata

LPRINT lista - scrie la imprimanta

MERGE "nume" - incarca in memorie un program de pe caseta, dar nu sterge in intregime pe cel vechi, ci inlocuieste numai liniile cu acelasi numar

NEW - sterge din memorie programul si variabilele

NEXT v - produce reluarea executiei unui ciclu FOR pentru valoarea urmatoare a variabilei v.

```
Ex.:FOR k=1 TO 10 STEP 2
    :
    :
    NEXT k
```

a OR b - daca b=0 a OR b = a
b<>0 a OR b = 1

OUT m,n - inscrie valoarea octetului n in portul de intrare/iesire cu adresa m

OVER n - daca n=1, produce supratiparirea unui nou caracter deasupra celui vechi, in pozitia in care se afla prompter-ul de linie

PAPER n - coloreaza fondul (hirtia) ecranului cu culoarea data de n (n=0 la 7)

```
Ex.:BORDER 5:PAPER 7
    LET bw=2:PAPER bw
```

PAUSE n - opreste executia programului pe o perioada de n/50 secunde (n=0-65535), ecranul raminind neschimbat. Cind se da PAUSE 0 reluarea programului se face la apasarea oricarei taste.

PEEK adresa - citeste din memorie si afiseaza octetul de la "adresa"
Ex.:PEEK 23627

```
FOR a=0 TO 20
PRINT a;TAB 10;PEEK a
NEXT a
```

PI - introduce valoarea 3,141592265

PLOT x,y - muta cursorul grafic in pozitia de coordonate x,y (x=0-255 si y=0-175) si desemneaza un punct.

PLOT m;x,y - m reprezinta INVERSE/OVER

Ex: PLOT INVERSE 1,30,40
PLOT OVER 1,100,100

POKE m,n - scrie valoarea n in octetul cu adresa m din memorie (m=0 la 65535 , n=0 la 255)

Ex: POKE 23692,255
POKE USR "M" + i

POINT (x,y) - da informatii despre punctul de coordonate (x,y);
.valoarea 1 - daca punctul este vizibil (culoarea cernelii)
.valoarea 0 daca punctul este stins (culoarea hirtiei).

PRINT... afiseaza pe ecran constante , variabile sau expresii ,separate in instructiune prin virgula,punct si virgula sau apostrof.

PRINT AT l,c;...- afiseaza incepind cu pozitia data de coordonatele l si c (linia =0-21 si coloana =0-31)

Ex: PRINT AT 11,13; "CENTRUL"
PRINT A$(x TO 6),A$(7-n) to 6

PRINT TAB C;...- afiseaza incepind cu coloana C (C=0-31).

Ex:PRINT AT 3,1;"Capitol";TAB24;"Pagina"

RND - da un numar pseudoaleator cuprins intre 0 si 1 sau cuprins in orice alt domeniu de valori,astfel:

x*RND	numere intre 0 si 8
a+x*RND	numere intre a si x

Ex: 8*RND	numere intre 0 si 8
2.1+0,9*RND	numere intre 2.1 si 3
5+INT(ND*20)	numere intregi intre 5 si 20

RANDOMIZE n - genereaza o valoare n de start pentru instructiunea RND (n=1-65535)

RANDOMIZE sau RANDOMIZE 0 da o valoare n de start in functie de timpul trecut de la pornirea calculatorului.

Ex: RANDOMIZE 1
FOR n=1 TO 10:PRINT RND
NEXT n

READ v1,v2...- atribuie variabilelor v1,v2...valoarea unor expresii succesive listele instructiunilor DATA.

Ex: READ A(x)
DATA 7,35,42,83
DATA "ART1","ART2".

ANEXA C

REM comentariu - permite introducerea de comentarii in program.

Ex: REM se executa desenul.

RESTORE n - restabileste indicatorul de citire in lista de instructiuni DATA cu numar de linie n, astfel incit urmatoarea instructiune READ va incepe citirea de la aceasta linie.

Ex:
```
 40 RESTORE 100
 50 READ a,b
 60 DATA 40,65,82,96
 :
100 DATA 26.7,"ABC",-52,75
```

RETURN - instructiune obligatorie de incheiere a unei subrutine si provoaca intoarcerea executiei programului la prima instructiune dupa GOSUB n

RUN - lanseaza in executie programul incepind cu prima linie.

RUN n - lanseaza in executie programul incepind cu linia n.

SAVE "nume" - copiaza programul din memorie pe caseta,cu numele indicat.

SAVE "nume" LINE n - inregistreaza programul pe caseta astfel incit la reincarcarea lui in memorie va fi lansat automat in executie incepind cu linia n.

SAVE "nume"DATA nume matrice - inregistreaza pe caseta un tablou.

SAVE "nume"CODE adresa,numar - copiaza pe banda un numar de octeti din mamorie incepind de la adresa specificata.

SCREEN $ (x,y) - da caracterul ASCII aflat in pozitie determinata de coordonatele x (linia) si y (coloana).

Ex: PRINT SCREEN$ (10,18)

SGN x - da semnul lui x:
-1 daca x<0
0 daca x=0
1 daca x>0

SIN x -calculeaza sinusul unghiului x (x in radiani)

Ex: SIN(n/6*PI)

SQR x - calculeaza radical de ordin doi din x (x>0)

Ex:
```
SQR 4
SQR 0.625
SQR (n/64)
```

STR$ x - converteste argumentul x intr-un sir

Ex: LET n$=STR$ 2E3

STOP - opreste executia programului afisindu-se mesajul:
STOP statement , n:m

Executia programului se reia cu CONTINUE

TAN x - calculeaza tangenta unghiului x (x in radiani).

USR adresa - lanseaza in executie o subrutina scrisa in cod masina memorata incepind cu adresa specificata.

Ex: INPUT "rind"
POKE USR "a"+n , rind

USR x$ - x este o litera intre a si u,sau un caracter grafic definit de utilizator.Se obtine adresa primului octet pentru caracterul grafic respectiv.

VAL "x" - evalueaza sirul x ca o expresie numerica.

Ex: VAL "3*5"=15

VAL$ "x" - evalueaza sirul x ca o expresie sir.

VERIFY "nume" - verifica daca inregistrarea de pe banda magnetica are acelasi continut cu memoria.

INCARCAREA SI COPIEREA PROGRAMELOR
ANEXA D

Dupa conectarea la retea, automatul programabil se afla sub controlul unui program rezident in memorie.

Acest program permite realizarea urmatoarelor operatii:

- incarcarea de pe banda a unui interpretor (ex:BASIC)
- informarea utilizatorului privind continutul unei casete cu programe
- incarcarea, salvarea si verificarea programelor, cu patru densitati de inmagazinare a datelor
- reglarea parametrilor casetofonului.

Utilizatorul este informat despre principalele optiuni, prin intermediul unui meniu:

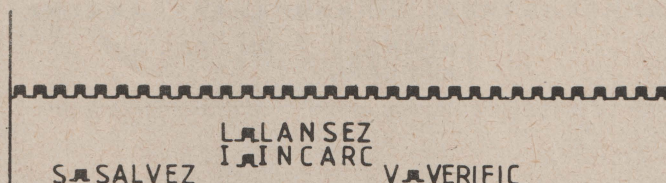


fig.D1

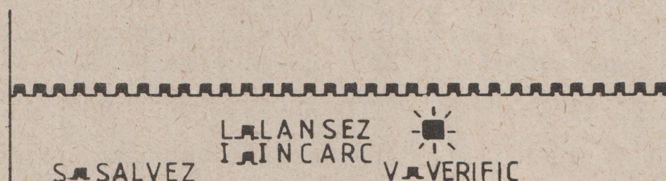
Literele prezente in fata optiunilor sugereaza tastele ce trebuie apasate pentru selectarea modurilor de lucru.

Apasarea unei taste este confirmata de prezenta unui semnal sonor in difuzor si aparitia cursorului clipitor in dreapta optiunii selectate.

L-LANSEZ

CIP-ul poate fi folosit pentru realizarea sau exploatarea unor programe scrise intr-un limbaj evoluat (ex:BASIC,PASCAL) si pentru rularea unor programe in cod masina.

Dupa apasarea tastei <L>, ecranul arata ca in fig.D2



Puteti continua in doua moduri:

1. - apasind <ENT>, va fi incarcat urmatorul program de pe banda.
2. - tastind litera cu litera numele programului pe care il asteptati de pe banda si apoi <ENT>.

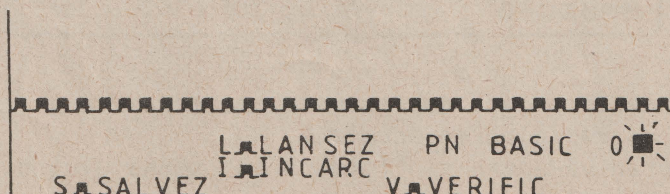


fig.D3

OBSERVATII:

1. Daca tastati in mod eronat un nume, puteti reveni in meniul principal cu <BREAK>, pentru a relua operatia.

2. CIP-ul poate selecta si incarca un program de pe banda chiar daca se tasteaza numai primele caractere din nume.

Dupa <ENT>, sau dupa tastarea a mai mult de zece caractere, va apare mesajul clipitor "CAS" care va sugereaza sa porniti casetofonul pe redare si sa apasati o tasta oarecare (cu exceptia tastei <BREAK>, care semnifica abandonarea optiunii si intoarcerea in meniul principal).

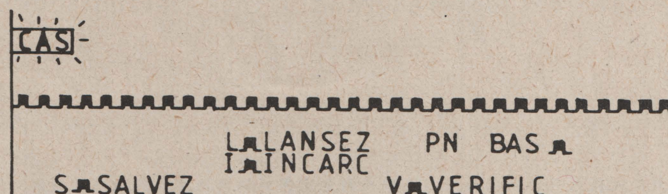


fig.D4

Se va porni casetofonul si se va astepta incarcarea programului dorit.

Un program inregistrat pe banda are un "antet", o "carte de vizita", ce constituie primul lucru incarcat in memorie si afisat pe ecran.

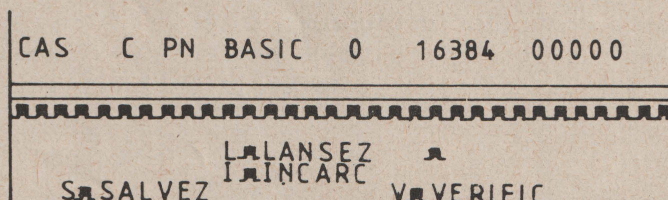


fig.D5

Informatia afisata consta din:

- tipul programului (B,C,D): B=BASIC, C=COD MASINA, D=DATE
- numele programului
- lungimea programului (in numar de octeti)
- adresa la care a fost salvat (pentru programe cod masina) sau linia de autostart (pentru programe BASIC)

Apoi CIP-ul va incarca programul propriu-zis in memorie, la adresa 00000.

Daca operatia s-a finalizat cu succes, programul incarcat va prelua controlul aparatului.

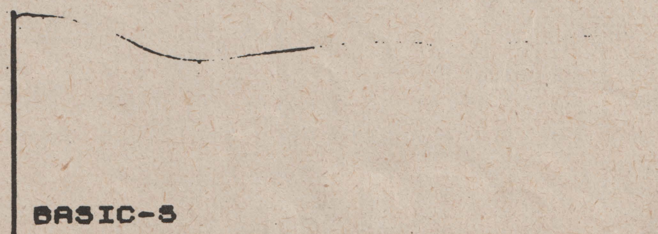


fig.D6

Daca operatia de incarcare nu reuseste, pe ecran va apare scris cuvintul "ERONAT", fiind necesara reluarea acesteia.

Totodata pe ecran va apare si "fotografia" semnalului distorsionat care a dus la oprirea incarcarii.

OBSERVATIE:

Oprirea casetofonului in diverse faze ale incarcarii unui program va genera de asemenea pe ecran mesajul "ERONAT".

Programul rezident ofera si posibilitatea copierii cu patru trepte de densitate a programelor de orice tip, de pe o caseta pe alta prin intermediul CIP-ului.

Aceasta facilitate poate fi folosita pentru organizarea si condensarea bibliotecii dumneavoastra de programe.

Transferul de date de pe o caseta pe alta se face in trei etape in ordinea: INCARC, SALVEZ, VERIFIC.

I-INCARC

Pentru inceput trebuie sa incarcati programele pe care doriti sa le copiati, de pe caseta origine, in memoria RAM a calculatorului. Aveti la dispozitie pentru aceasta 58100 octeti de memorie, in maximum 15 tronsoane.

Apasati <I>. Va reamintim ca exista urmatoarele alternative:

- tastati <ENT> pentru a incarca urmatorul program de pe banda
- tastati numele programului (intreg sau partial) si apoi <ENT>

Mesajul "CAS" clipeste, solicitindu-va sa porniti casetofonul (in care se afla pozitionata caseta origine) si sa apasati orice tasta, in afara de <BREAK>.

Pe ecran va fi afisata de acum inainte, permanent, capacitatea de memorie disponibila pentru stocarea programelor in vederea copierii lor pe alte casete.

Incarcarea debuteaza cu scrierea pe ecran a "antetului" programului (header).

Daca incarcarea tronsonului se finalizeaza cu succes,CIP-ul afiseaza antetul programului in partea de sus a ecranului si contorul memoriei disponibile scade cu lungimea corespunzatoare programului incarcat (antetul ocupa 17 octeti, asa ca pentru fiecare program cu header, incarcat, se scade 17 din contorul memoriei disponibile).

De obicei tronsoanele programelor exitente au capacitati mari si sint suficiente doar citeva pentru coborirea contorului memoriei disponibile spre 00000.

In situatia in care lungimea unui program este mai mare decit capacitatea de memorie ramasa dsponibila, se revine automat in meniul principal, afisindu-se mesajul "DEPASIT".

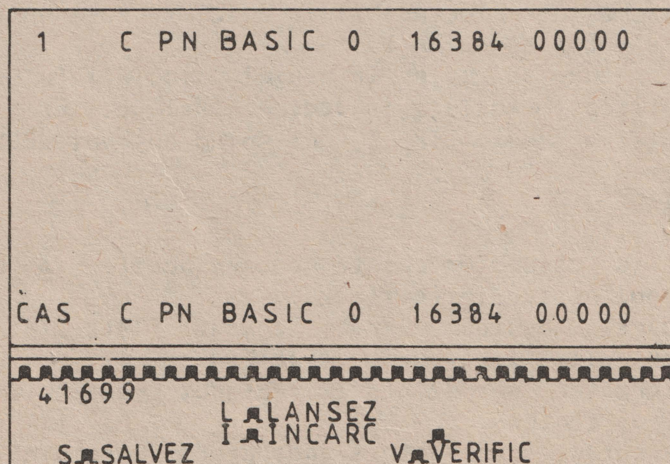


fig.D7

```
1      C LOAD 1 4          06808 45074
2      C ROMLOAD           16384 00000
3      B P                 00054 00010
4      C 2                 04694 45100
5      C J                 10536 55000



       B J                 27403 00040

DEPASIT
 19589

           LANSEZ
           INCARC

  SALVEZ              VERIFIC
```

fig.D8

OBSERVATIE:

Exista si programe care nu au "antet" (se numesc headerless-uri).

In acest caz pe ecran vor fi afisate: un oblon (semn special) si, (la terminarea incarcarii), lungimea tronsonului de program incarcat.

La incarcarea unui program fara antet in cazul in care lungimea lui depaseste memoria libera existenta (avind in vedere ca ea nu poate fi apreciata de la bun inceput), calculatorul va incarca datele pina la epuizarea totala a spatiului disponibil, moment in care va reveni in meniul principal afisind pe ecran mesajul "ERONAT", fara a lua in considerare incarcarea nereusita.

Se revine in meniul principal si in cazul in care numarul tronsoanelor incarcate depaseste -15 (indicator F).

Din aceasta optiune se revine in meniul principal cu <BREAK>.

Pentru "curatirea" memoriei, in vederea unei noi serii de incarcari, se va tasta <A>, sau se va reseta CIP-ul, actionind butonul RESET.

S-SALVEZ

Introduceti in casetofon caseta pe care doriti sa copiati programul pregatiti casetofonul pentru inregistrare.

Dupa efectuarea acestor operatii apasati tasta <S> si la aparitia cursorului clipitor puteti tasta doua caractere:

- primul caracter reprezinta indicatorul tronsonului de la care se declanseaza lantul de copiere.
- al doilea caracter reprezinta indicatorul ultimului tronson ce va fi copiat.

OBSERVATIE:

Cererile incorecte nu sint executate, producindu-se o intoarcere la meniul principal.

Operatia de salvare propriu-zisa este demarata de o noua tastare, ce urmareste sa stabileasca densitatea de salvare (de la 1 la 4) a datelor pe banda.

Apasind orice tasta cu exceptia tastelor <2>,<3>,<4>, salvarea se va face cu densitate 1 (densitate normala).

Apasind una din tastele <2>,<3>,<4>, salvarea se va face cu densitatea corespunzatoare cifrei.

Fiecare tronson salvat este marcat pe ecran.

O data aleasa, densitatea ramine aceeasi pentru toate programele, dar poate fi modificata "din mers", apasind pe cifra corespunzatoare la inceputul unui nou tronson.

Folosirea densitatilor superioare celei normale pentru salvarea programelor, va permite sa faceti economie de timp (la incarcare/salvare) si de casete.

Acei dintre dumneavoastra care poseda aparate corect reglate, cu performante bune, vor putea face salvari fara probleme, cu orice densitate.

Pentru densitati mari (3,4), va recomandam sa utilizati acelasi casetofon si pentru salvarea si pentru incarcarea programelor.

Efectuarea unor copii de calitate cu casetofoane ce au posibilitatea reglarii nivelului la inregistrare, impune pozitionarea acului (sau a LED-urilor) indicatoare pe inceputul zonei rosii (deci, nivel mai mare decit cel admis la copierea muzicii).

De asemenea este bine sa comutam casetofoanele pe optiunea "MONO", la incarcarea programelor.

Operatiile de salvare si incarcare pot fi stopate oricind in timpul desfasurarii lor prin apasarea tastei <BREAK>.

V-VERIFIC

Pentru a fi siguri ca datele inregistrate pe banda sint identice cu cele din memoria calculatorului, e necesar sa efectuati verificarea programelor salvate.

In acest scop veti pozitiona caseta pe inceputul inregistrarilor facute si veti declansa operatia de verificare secventa cu secventa.

Apasati tasta <V>. La fel ca la salvare, aveti posibilitatea sa tastati numarul de ordine al tronsonului pe care doriti sa-l verificati sau cu care vreti sa inceapa seria verificarilor si al tronsonului final.

Tronsoanele verificate vor fi marcate pe ecran.

Verificarea se desfasoara asemanator cu o incarcare de program obisnuit.

Daca operatia este finalizata cu succes, pe ecran apare mesajul "BUN" si apoi calculatorul trece la verificarea programului urmator din tabel; in caz contrar se intoarce in meniul principal cu mesajul "ERONAT".

OBSERVATII:

1.In situatia in care nu veti pozitiona caseta corect pentru verificarea programului copiat, pe ecran vor fi afisate toate antetele citite de pe banda, dar operatia de verificare va incepe numai dupa intilnirea antetului cautat.

In cazul unui program fara antet nepotrivirea duce la aparitia masajului "ERONAT" si la revenirea in meniul principal.

Daca v-ati pozitionat corect si totusi aveti eroare la verificare, trebuie sa repetati operatia de salvare.

2.La operatiile de incarcare si verificare nu este necesar sa precizati densitatea cu care a fost inregistrat programul, deoarece CIP-ul o va recunoaste singur.

3.Desi nu constituie o optiune destinata, puteti cunoaste rapid continutul unei casete inregistrate sau cauta un anumit program prin intermediul tastelor <L> sau <I>.

Dupa apasarea uneia dintre ele, scrieti un nume oarecare compus din unu pina la zece caractere, care sa nu coincida cu nici un alt nume de program de

pe banda apoi tastati <ENT>.

CIP-ul va publica pe ecran informatii despre programele existente pe banda, fara a le incarca in memorie.

Dupa pornirea casetofonului si <ENT>, aparatul va intra intr-o bucla de citire, din care poate fi scos apasind tasta <BREAK>.

4.Intre doua titluri de programe citite de pe banda, in partea de jos a ecranului apar "fotografii" ale formei semnalului. Aceste imagini va ajuta sa depistati cauza unei eventuale erori de incarcare (amanunte la punctul 6).

5.La citirea antetelor programul rezident nu afiseaza pe ecran decit literele si cifrele din numele unui program. Restul caracterelor ASCII sint inlocuite printr-un "oblon" negru. La copiere se conserva insa toate caracterele ASCII.

6.REGLAJUL CASETOFONULUI

Desi nu este specificata in meniul principal (adresinu-se specialistilor), aceasta optiune se selecteaza apasind tasta <R>.

Porniti casetofonul, tastati <ENT>, si veti putea urmari pe ecran imaginea dinamica a semnalului inregistrat pe banda.

Cu ajutorul tastei <S> puteti realiza un "stop cadru".

Iesirea din "stop cadru", se realizeaza apasind orice tasta.

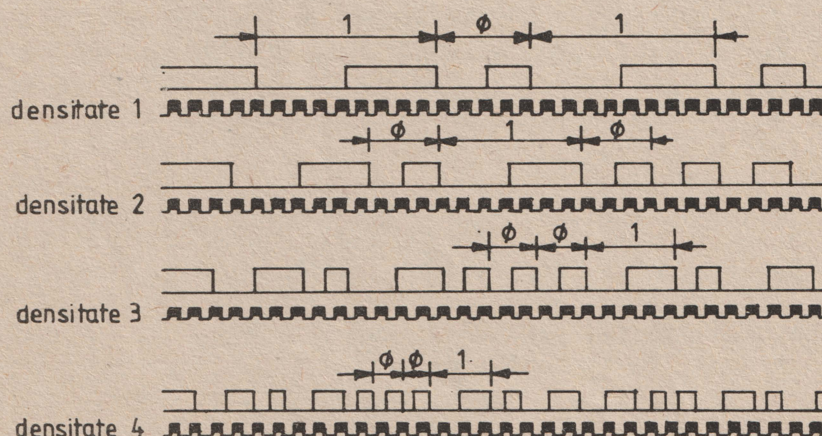
"Fotografii" repetate va vor permite sa va edificati asupra calitatii inregistrarii sau a posibilitatilor casetofonului folosit. Cu tasta <BREAK> reveniti in meniul principal.

Este posibil sa nu va reuseasca incarcarea unor programe.

In acest caz e bine sa va adresati specialistului pentru depanarea sau reglarea aparatului.

Exista cazuri in care inregistrari facute pe alte casetofoane sau pe anumite tipuri de casete, produc eroare la incarcare. In cazul in care casetofonul utilizat dispune de reglaj de volum si ton al semmnalului injectat in calculator (de exemplu la iesirea de difuzor sau de casca) sau aveti un amplificator ori un egalizor ce va permite sa interveniti asupra nivelului si caracteristicii de frecventa a semnalului generat de casetofon, va oferim un ghid cuprinzind exemple de alterare a inregistrarii si indicatii specifice de reglaj.

1.Semnal normal (fig.D9)

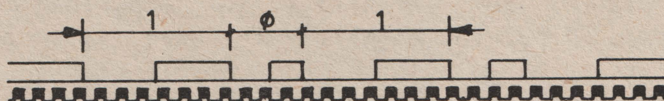


Puteti urmari semnale cu densitate 1,2,3,4 (fig.D8 - semnale continind biti "1" si "0" cu patru densitati).

Observati ca alternantele semnalului pastreaza aceeasi forma, avind densitatile:

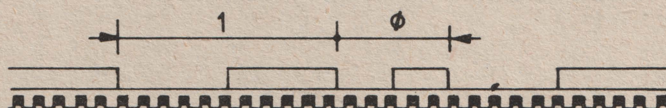
- 1 : 100%
- 2 : 66%
- 3 : 50%
- 4 : 33%, raportate la densitatea standard.

2. Turatie mai mare a casetofonului (fig.D10)



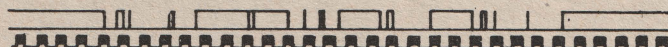
REGLAJ: Se va reduce turatia casetofonului pina cind semnalul va avea forma corecta.

3. Turatie mai mica a casetofonului (fig.D11)



REGLAJ: Se va mari turatia casetofonului.

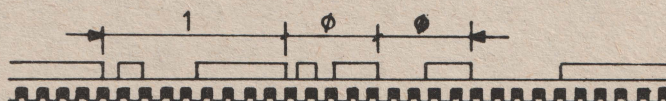
4. Semnal cu zgomote (fig.D12)



REGLAJ: Se va actiona asupra volumului si tonului pina la disparitia completa a zgomotului.

In cazul in care nu dispare depanati casetofonul.

5. Semnal cu nivel prea mare (cu alternante suplimentare) (fig.D13)



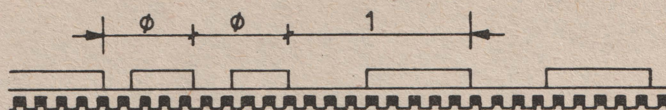
REGLAJ: Din volum se reduce nivelul semnalului pina la obtinerea unei forme corecte.

6. Semnal cu nivel prea mic (cu alternante pierdute) (fig.D14)



REGLAJ:Din volum se va mari nivelul semnalului.

7. Semnal asimetric (fig.D15)



REGLAJ: Se va actiona asupra tonului si volumului pina la obtinerea unui semnal corect. In caz de nereusita se va modifica pozitia capului de redare/inregistrare pina in momentul in care alternantele vor deveni simetrice.

8. Lipsa semnal pe banda (fig.D16)



ANEXA E
Facilitati suplimentare de lucru ale interpretorului "PN BASIC 0"

Acest program aflat pe caseta demonstrativa este un interpretor BASIC - modificat, compatibil cu interpretorul BASIC - SPECTRUM SINCLAIR si care va ofera noi facilitati:

1. Posibilitatea conlucrarii interpretorului BASIC cu programul rezident. Astfel, pe linga rutinele standard din BASIC - SPECTRUM, aveti posibilitatea oricind sa urmariti reglarea casetofonului, sa va informati asupra continutului unei casete sau sa copiati programe pe patru densitati.

Pentru aceasta se tasteaza <CS+SS> (modul E), apoi <SS+CR>.

Pe ecranul televizorului se afiseaza meniul principal al programului rezident.

O singura deosebire: memoria disponibila pentru copiere va fi in acest caz diminuata cu lungimea programului BASIC existent in acel moment in calculator.

Se revine in interpretorul BASIC tastind <A> din meniul principal al programului rezident.

Aveti deci la dispozitie un interpretor de 18K (16K+2K).

2. Utilizatorii incepatori, tastind <CS+SS> urmat de <SS+BREAK>, pot obtine in paranteze, traducerea in limba romana a termenilor din limba engleza folositi de limbajul BASIC pentru denumirea instructiunilor si functiilor (fac exceptie cuvintele-cod netraductibile).

Se revine la varianta standard dupa aceeasi dubla tastare:
<CS+SS>, <SS+BREAK>.

3. Daca la un moment dat vi se defecteaza o tasta, tastati <CS+SS> urmat de <CS+SPACE> si veti obtine schimbarea semnificatiei tastelor claviaturii prin citirea lor "in oglinda". De exemplu:

<1> devine <0> si <0> devine <1>
<2> devine <9> si <9> devine <2>
....
<5> devine <6> si <6> devine <5>
....
<A> devine <ENT> si <ENT> devine <A>
....

Fac exceptie tastele: <CS>, <Z>, <SS>, <BREAK>, a caror semnificatie ramine aceeasi.

Se revine la situatia normala dupa o noua dubla tastare:
<CS+SS> urmat de <CS+SPACE>.

4. Rutinele LOAD, VERIFY, MERGE, contin perfectionari fata de varianta SPECTRUM si lucreaza cu patru densitati de inmagazinare a datelor pe banda, densitatea fiind recunoscuta automat.

5. Rutina SAVE poate copia (salva) programe cu patru densitati, functie de tasta apasata (<1>, <2>, <3>, <4>), dupa aparitia mesajului "1.Porniti cas. 2.Apasati o tasta"

6. Rutina USR are corectata greseala din varianta SPECTRUM

7. Mesajele de eroare sint afisate in limba romana.

Programul "PN BASIC 0" are si rolul de a va sublinia marele avantaj oferit in premiera de acest calculator, acela de a lucra cu mai multe variante de intrerpretoare.

Mai mult, "CIP-ul" va permite sa il destinati si altor limbaje sau sa construiti singuri facilitati suplimentare de lucru.

ANEXA F
VARIABILE DE SISTEM

Octetii din memorie de la adresa 23552 la adresa 23733 sint rezervati pentru operatii specifice ale sistemului. Ei pot fi cititi pentru a afla diferite lucruri despre sistem, iar citiva din ei pot fi si modificati. Acesti octeti se numesc variabile de sistem si au cite un nume, dar nu trebuie confundati cu variabilele utilizate de BASIC. In cazul variabilelor formate din mai multi octeti, primul va fi octetul cel mai putin semnificativ. Variabilele de sistem sint date in lista de mai jos. Abrevierile din coloana 1 au urmatoarea semnificatie:

X - aceasta variabila nu poate fi modificata deoarece sistemul va functiona eronat

N - modificarea acestei variabile nu are un efect asupra functionarii normale a sistemului

n - numarul de octeti din variabila

Tip	Adresa	Nume	Continut
N8	23552	KSTATE	Folosita in citirea tastaturii
N1	23560	LAST K	Retine ultima tasta apasata
1	23561	REPDEL	Durata (in 1/50 sec.) cit trebuie tinuta apasata tasta pentru a se repeta, initial 35
1	23562	REPPER	Timpul (in 1/50 sec.) dupa care se repeta o tasta apasata, initial 5
N2	23563	DEFADD	Adresa argumentelor functiilor definite de utilizator daca una dintre ele a fost evaluata
N1	23565	K DATA	Al doilea octet pentru controlul culorii introdus de la tastatura
N2	23566	TVDATA	Controlul culorii, al lui AT si TAB pentru TV
X38	23568	STRMS	Adresa canalului atasat caii
2	23606	CHARS	Adresa generatorului de caractere minus 256 in mod normal in BASIC-S, dar va puteti genera unul si dvs. facind ca CHARS sa pointeze la el
1	23608	RASP	Durata sunetului de atentionare
1	23609	PIP	Durata sunetului la apasarea unei taste
1	23610	ERR NR	Codul de mesaj minus 1
X1	23611	FLAGS	Diferiti indicatori de control ai sistemului BASIC
X1	23612	TVFLAG	Indicatorii asociati cu TV-ul
X2	23613	ERR SP	Adresa elementului din stiva masinii utilizat ca adresa de introducere in caz de eroare
N2	23615	LIST SP	Adresa de intoarcere la listarile automate
N1	23617	MODE	Specifica cursorul (K,L,C,E,G)
2	23618	NEWPPC	Linia la care se sare
1	23620	NSPPC	Numarul instructiunii in linie la care se sare
2	23621	PPC	Numarul liniei pentru instructiunea in executie
1	23623	SUBPPC	Numarul instructiunii din linie in executie
1	23624	BORDER	Culoarea border-ului inmultita cu 8; contine de asemenea atributele folosite pentru partea de jos a ecranului
2	23625	E PPC	Numarul liniei curente
X2	23627	VARS	Adresa variabilelor
N2	23629	DEST	Adresa variabilelor asignate
X2	23631	CHANS	Adresa datelor de canal
X2	23633	CURCHL	Adresa informatiei curente folosita pentru intrare sau iesire
X2	23635	PROG	Adresa programului BASIC
X2	23637	NXTLIN	Adresa urmatoarei linii de program
X2	23639	DATADD	Adresa ultimului element din lista DATA
X2	23641	E LINE	Adresa comenzii introduse

2	23643	K CUR	Adresa cursorului
X2	23645	CH ADD	Adresa urmatorului caracter care urmeaza sa fie interpretat
2	23647	XPTR	Adresa caracterului dupa semnul intrebarii
X2	23649	WORKSP	Adresa spatiului de lucru temporar
X2	23651	STKBOT	Adresa inferioara a stivei calculator
X2	23653	STKEND	Adresa de inceput a spatiului liber
N1	23655	BREG	Registrul B al calculatorului
N2	23656	MEM	Adresa spatiului folosit pentru memoria calculatorului
1	23658	FLAGS2	Alti indicatori
X1	23659	DF SZ	Numarul liniilor din partea de jos a ecranului
2	23660	S TOP	Numarul liniei de sus a programului la listare automata
2	23662	OLDPPC	Numarul liniei la care sare CONTINUE
1	23664	OSPPC	Numarul din linie la care sare CONTINUE
N1	23665	FLAGX	Diversi indicatori
N2	23666	STRLEN	Lungimea asignata sirului
N2	23688	T ADDR	Adresa urmatorului element din tabela sintaxa
2	23670	SEED	Variabila pentru RND
3	23672	FRAMES	Controlul de cadre pe 3 octeti, cel mai putin semnificativ primul, incrementat la 20 ms
2	23675	UDG	Adresa primului grafic definit de utilizator
1	23677	COORDS	Coordonata x a ultimului punct plot-at
1	23678		Coordonata y a ultimului punct plot-at
1	23679	P POSN	Numarul pozitiei de scriere pe ecran
1	23680	PR CC	Octetul mai putin semnificativ al adresei pentru noua pozitie la care se imprima prin LPRINT
1	23681		Nefolosit
2	23682	ECHO E	Numarul coloanei si al liniei
2	23684	DF CC	Adresa de afisare pe ecran prin PRINT
2	23686	DFCCL	Acelasi lucru pentru partea de jos a ecranului
X1	23688	S POSN	Numarul coloanei pentru PRINT
X1	23689		Numarul liniei pentru PRINT
X2	23690	SPOSNL	Ca S POSN pentru partea de jos a ecranului
1	23692	SCR CT	Numara defilarile de ecran
1	23693	ATTR P	Culoarea curenta
1	23694	MASK P	Folosit pentru culori transparente
N1	23695	ATTR T	Culori temporare
N1	23696	MASK T	Ca MASK P dar temporar
1	23697	PFLAG	Alti indicatori
N30	23698	MEMBOT	Arie memorie calculator
2	23728		Nefolosit
2	23730	RAMTOP	Adresa ultimului octet din aria sistemului BASIC
2	23732	P-RAMT	Adresa ultimului octet de RAM

LISTA SEMNALELOR PREZENTE LA CONECTORUL DE EXTENSIE

Fata A	Nr.	Fata B
A15	1	A14
A13	2	A12
D7	3	VCC (+5V)
NC	4	NC
SLOT	5	SLOT
D0	6	GND
D1	7	GND
D2	8	CLK (3.5 MHz)
D6	9	A0
D5	10	A1
D3	11	A2
D4	12	A3
INT	13	IORGE
NMI	14	GND
HALT	15	VIDEO
MREQ	16	NC
IORQ	17	NC
RD	18	NC
WR	19	BUSRQ
NC	20	RESET
WAIT	21	A7
NC	22	A6
NC	23	A5
M1	24	A4
RFSH	25	ROMINH
A8	26	BUSACK
A10	27	A9
VINH	28	A11

OBSERVATII:

1. Fata A este fata plantata .
2. Semnalele sint numerotate privind dinspre fata A .de la stinga la dreapta ,tinind calculatorul in pozitie normala de lucru.
3. Asignarea semnalelor la conectorul de extensie este identica cu cea de la calculatorul SINCLAIR SPECTRUM ,exceptind tensiunile de -5V,+12V,-12V si semnalele U,V,Y (semnale de culoare) care nu se regasesc la calculatorul CIP .

I. P. "Filaret" Str. F-ca de Chibrituri nr. 9-11
cda. 860/90